Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина
Уральский энергетический институт

О. М. КОТОВ
Е. Н. КОТОВА
А. М. ВЕРХОЗИН

ОСНОВЫ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ДАННЫХ В ЦИФРОВЫХ СИСТЕМАХ

Учебное пособие

Министерство науки и высшего образования Российской федерации
Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина

О. М. Котов, Е. Н. Котова, А. М. Верхозин

ОСНОВЫ ПРЕДСТАВЛЕНИЯ И ОБРАБОТКИ ДАННЫХ В ЦИФРОВЫХ СИСТЕМАХ

Учебное пособие

Рекомендовано методическим советом
Уральского федерального университета
для студентов вуза, обучающихся
по направлению подготовки
13.04.02 — Электроэнергетика и электротехника

2-е издание, переработанное
и дополненное

Екатеринбург
Издательство Уральского университета
2020

УДК 004:621.37(075.8)
ББК 32-013я73
К73

Рецензенты:
кафедра автоматизированных электроэнергетических систем Самарского государственного технического университета (завкафедрой д-р техн. наук, проф. *В. П. Степанов*);
канд. техн. наук *В. Г. Неуймин* (начальник Центра моделирования и автоматизации управления энергосистем АО «Научно-технический центр Единой энергетической системы»)

Научный редактор — канд. техн. наук, доц. *П. А. Крючков*

Котов, О. М.
К73 Основы представления и обработки данных в цифровых системах : учебное пособие / О. М. Котов, Е. Н. Котова, А. М. Верхозин ; Мин-во науки и высш. образования РФ. — 2-е изд., перераб. и доп. — Екатеринбург : Изд-во Урал. ун-та, 2020. — 208 с.

ISBN 978-5-7996-3012-6

Настоящее учебное пособие содержит теоретический материал, примеры с решениями, задания для самостоятельных упражнений, а также рекомендации для практического освоения технологий обработки данных с использованием универсального программного обеспечения. Раздел 14 и приложение 2 подготовлены А. М. Верхозиным, разделы 10–13 — Е. Н. Котовой, остальные разделы подготовлены О. М. Котовым.

Библиогр.: 8 назв. Табл. 31. Рис. 148. Прил. 2.

УДК 004:621.37(075.8)
ББК 32-013я73

ISBN 978-5-7996-3012-6

Оглавление

Введение

Информатика является наукой, которая изучает общие свойства и структуру информации, закономерности и принципы ее создания, преобразования, накопления, передачи и использования в различных сферах человеческой жизнедеятельности.

Информация (от латинского *informatio* — разъяснение, изложение) — это набор сведений, передаваемых в различных системах, начиная от растительного мира и заканчивая человеческим обществом в целом.

С точки зрения математики, информация — это количественная мера устранения (или, по крайней мере, снижения) неопределенности сведений (представления) о некотором объекте. Наименьшее сообщение, снижающее неопределенность о чем-либо или о ком-либо, можно себе представить, как один из двух вариантов ответа, образующих полный набор (да/нет, имеется/отсутствует и т. п.). Объем подобного сообщения соответствует минимальной порции информации и называется бит.

Информация материализуется в виде сигналов. В технических системах под сигналом понимается изменяющийся во времени *физический процесс, параметры* которого могут принимать технически *различимые* значения [1].

При использовании всей совокупности значений параметра в некотором диапазоне говорят об аналоговом сигнале, если же речь идет о фиксированных (разрешенных) значениях — о дискретном или цифровом сигнале.

Универсальным техническим средством обработки информации является компьютер. Настоящее учебное пособие содержит минимально необходимые сведения о принципах работы компьютера, дает представление о современном состоянии и направлении развития компьютерной техники и программных средств, позволяет приобрести начальные навыки работы при решении практических задач.

Слово «компьютер» происходит от латинского глагола «*computo*» — считать, вычислять. В английском языке, из которого заимствовано это слово, «*computer*» буквально значит «вычислитель». Как технический термин слово «компьютер» приобрело более широкий смысл: *электронное устройство для программируемой обработки данных*.

Обратимся к истории. Вместе с первыми признаками разделения труда у человека возникла необходимость в счетных устройствах. Для несложных операций зачастую было достаточно пальцев рук. Но потребность в вычислениях выходила за рамки возможностей подобного «калькулятора». Из всех устройств и приспособлений, используемых для вычислений, наиболее практичными оказались счеты. Изобретены они были более 1500 лет назад.

Только в XVII веке появились счетные устройства, составившие некоторую конкуренцию счетам. Прежде всего это устройство шотландского ученого Джона Непера, который дал определение логарифму и, используя свойство логарифма произведения двух чисел (который равен сумме логарифмов сомножителей), создал *механическое устройство* для умножения чисел.

Следующим этапом развития механических счетных устройств можно считать *суммирующую машину* французского ученого и изобретателя Паскаля. Блез Паскаль, сын сборщика налогов, прожил всего 39 лет, но вошел в историю как выдающийся математик, физик, писатель и философ. На мысль о создании суммирующей машины его навело наблюдение за бесконечными и утомительными расчетами своего отца. За десятилетие Блез Паскаль построил более 50 различных вариантов суммирующей машины, названной им *Паскалиной* (рис. I).

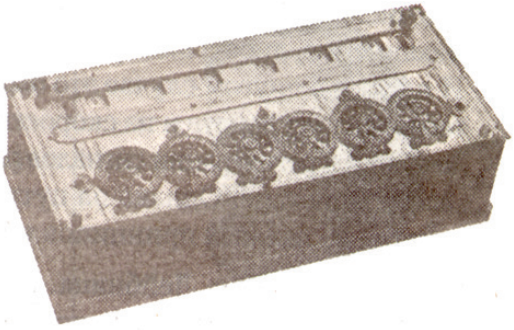


Рис. I. Суммирующая машина Паскаля

Суммирующая машина Паскаля представляла собой ящик с многочисленными шестеренками. Складываемые числа вводились путем соответствующего поворота наборных колес. Каждое колесо с нанесенными на нем делениями от 0 до 9 соответствовало одному разряду десятичного числа. Избыток над значением 9 колесо переносило на более «старшее» колесо (располагавшееся левее), поворачивая его на одно деление вперед. Остальные арифметические операции выполнялись на Паскалине при помощи довольно неудобной процедуры повторных сложений.

Первая машина, позволявшая, наряду со сложением, легко производить операции вычитания, умножения и деления была изобретена позже в том же XVII веке в Германии. Заслуга этого изобретения принадлежит Готфриду Вильгельму Лейбницу. Решение изобрести механическое устройство для арифметических расчетов было принято им после знакомства с голландским математиком и астрономом Христианом Гюйгенсом, которому приходилось регулярно выполнять вручную громоздкие и трудоемкие вычисления.

В 1673 году Лейбниц изготовил первый *механический калькулятор*. Этот механизм сделал автоматическими повторения, необходимые для умножения и деления чисел. Однако прославился Лейбниц прежде всего не этой машиной, а созданием дифференциального и интегрального счисления (параллельно с Исааком Ньютоном) и системы двоичного счисления, которая позже нашла применение в цифровых устройствах.

Из всех изобретателей прошлых столетий ближе всего к созданию компьютера подошел английский ученый Чарльз Бэббидж. В 1833 году он представил проект *аналитической машины*, которая должна была выполнять разнообразные вычислительные операции в соответствии с инструкциями, задаваемыми оператором. В действительности это был первый универсальный компьютер. Аналитическая машина (рис. II) имела такие компоненты, как «контора», «мельница» и «склад» (в современной терминологии — устройство управления, арифметико-логическое устройство (АЛУ) и память), состоящие из механических рычажков, шестеренок, передач, защелок, часовых механизмов и пр. Память вмещала 100 сорокоразрядных чисел. Предварительно размещенные в памяти эти числа оставались в ней до тех пор, пока не были востребованы для вычислений в «мельнице». Результаты операций либо отправлялись обратно в память, либо распечатывались. Инструк-

ции (команды), определяющие последовательность действий, вводились в аналитическую машину с помощью набора (колоды) перфокарт и анализировались (дешифровывались) в специальном устройстве под названием «контора».

Аналитическая машина, которая в своем окончательном виде оказалась размером не менее железнодорожного локомотива, состояла из более чем пятидесяти тысяч стальных, медных и деревянных механических деталей и приводилась в движение паровой машиной. Однако запустить ее в работу при жизни автора не удалось. В 1910 году она была все-таки запущена в работу и в демонстрационных целях выполнила расчет таблицы степеней числа π с точностью до 20 десятичных разрядов.

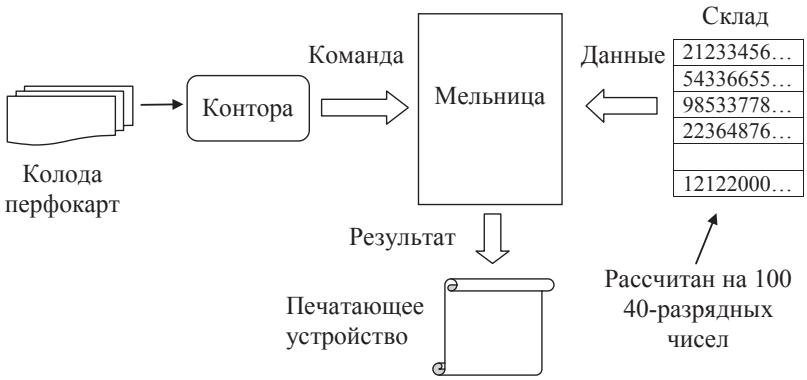


Рис. II. Схема Аналитической машины Чарльза Бэббиджа

Изобретатели компьютеров в первой половине двадцатого столетия получили в свое распоряжение электромеханические реле, электронные лампы и другие электротехнические компоненты. Первый в мире *немеханический компьютер*, созданный в 1939 году Джорджем Атанасовым, профессором физики колледжа штата Айова, имел арифметико-логическое устройство, выполненное на электронных лампах, память на конденсаторах, оперировал данными в двоичном представлении и назывался *ABC*. Компьютер на электромеханических реле был запущен Говардом Айкеном чуть позже в 1943 году и назывался *Марк I*. Во время Второй мировой войны он использовался для расчетов баллистических таблиц береговой артиллерии США. Для расшифровки секретных кодов немецких радиограмм в том же 1943 году в Англии в секретной лаборатории «Блечли–парк» был создан специализиро-

ванный компьютер *Colossus*, содержащий около двух тысяч электронных ламп.

Спустя всего четыре года после изобретения (в 1951 г.) плоскостного транзистора фирма «Белл телефон лабораториес» начала выпуск первого полностью транзисторного компьютера. Он назывался *Традис* и содержал примерно 800 транзисторов.

Толчком для следующего этапа развития компьютерной техники послужило изобретение в 1962 году интегральной микросхемы. Она содержала первоначально только 10 транзисторов, но их количество (интеграция) стремительно и неуклонно повышалось. И уже в 1971 году Эдвардом Хоффом был изготовлен первый микропроцессор **Intel 4004**, имевший в своем составе 2250 элементов. А на базе микропроцессора **Intel 8008** (4500 элементов), созданного в 1972 году, Эдвард Робертс в 1974 году собрал первый в мире персональный компьютер *Альтаир*.

1. Внутреннее представление информации

Системы счисления

В значительной степени характеристики любого вычислительного устройства зависят от того, какая система счисления положена в основу его работы.

Все используемые в настоящее время системы счисления можно разделить на две большие группы: *позиционные* и *непозиционные*. К позиционным относятся те системы, вес цифры в которых зависит от места (или позиции) этой цифры в числе. В непозиционных системах нет подобной закономерности: там значение цифры неизменно и не зависит от ее места в числе. Из непозиционных до наших дней сохранилась только *римская система счисления*. Например, запись M*DLCXIV* в этой системе означает ни что иное, как число 1000 + 500 – 50 + 100 + + 10 – 1 + 5 = 1564. В этой системе *I* — это единица, *V* — пять, *X* — десять, *L* — пятьдесят, *C* — сто, *D* — пятьсот, *M* — тысяча.

Гораздо более многочисленную группу образуют позиционные системы счисления [1]. Из всего многообразия позиционных систем наибольшее распространение получили правосмещенные канонические системы счисления. В таких системах используются только цифры, соответствующие значениям положительной полуоси целых чисел, натуральные основания и естественный порядок весов (0, 1, 2...). Об этих системах и пойдет речь в последующем изложении.

Различают две основные характеристики системы счисления: *основание* и *алфавит*. Основанием системы счисления служит число, степень которого является *«весом»* цифры в значении числа. От основания образуется и название системы счисления. Номер позиции цифры в числе является показателем степени основания. Позиции нумеру-

ются с нуля справа налево. Алфавит системы счисления — это набор цифр, с помощью которого записывается число. Этот набор всегда начинается с нуля и заканчивается цифрой, предшествующей основанию (табл. 1).

Таблица 1

Система счисления	Основание	Алфавит системы счисления
Двоичная	2	0,1
Восьмеричная	8	0,1,2,3,4,5,6,7
Десятичная	10	0,1,2,3,4,5,6,7,8,9
Шестнадцатеричная	16	0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F

Для любой позиционной системы счисления справедлива *формула значения числа*

$$\left(a_{n-1}a_{n-2}\ldots a_1a_{0,}a_{-1}a_{-2}\ldots a_{-m}\right)_b =$$

$$= a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \ldots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + a_{-2}b^{-2} + \ldots + a_{-m}b^{-m}, \qquad (1)$$

где b — основание системы счисления, a_i — цифра из алфавита системы счисления ($0 \le a_i \le b$).

Примечательно, что произнося числительные, мы по сути дела проговариваем формулу (1). Система счисления указывается обычно подстрочным индексом. Если индекс отсутствует, считается, что число задано в десятичной системе.

Примеры

$1234 = 1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^{0.}$

$1234_8 = 1 \cdot 8^3 + 2 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 = 512 + 128 + 24 + 4 = 668.$

$1234_{16} = 1 \cdot 16^3 + 2 \cdot 16^2 + 3 \cdot 16^1 + 4 \cdot 16^0 = 4660.$

$00111010{,}11_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} =$
$= 32 + 16 + 8 + 2 + 0{,}5 + 0{,}25 = 58{,}75.$

При решении подобных примеров удобно пользоваться таблицей степеней основания (табл. 2).

Таблица 2

n	7	6	5	4	3	2	1	0
2^n	128	64	32	16	8	4	2	1
8^n	2 097 152	262 144	32 768	4 096	512	64	8	1
16^n	268 435 456	167 777 216	1 048 576	65 536	4 096	256	16	1

В первую очередь преимущества позиционных систем счисления проявляются при выполнении арифметических операций.

Пример. Рассчитайте результат выражения $37_8 + 153_8 = ?_8$.

Вычисления удобнее производить «в столбик». В младшем разряде получаем 7 + 3 = 10. Поскольку слагаемые заданы в 8-й системе и 10 > 8 (8 — это основание системы счисления) на 2, двойка остается в текущей позиции, а 8 — в виде единицы переносится в старший разряд, так как более старший разряд домножается на основание 8 в степени на единицу большей, чем в рассматриваемом разряде. Аналогично рассчитываются результаты в остальных разрядах:

$$\begin{array}{r} 37_8 \\ \underline{+153_8} \\ 212_8 \end{array}$$

Ответ: 212_8.

Пример. Рассчитайте результат выражения $A7_{16} + 1EF_{16} = ?_{16}$.

Решение. Рассуждаем следующим образом. В нулевом разряде имеем $7_{16}+F_{16}=7+15=22$. Полученная сумма превосходит основание 16 на 6. Эта цифра остается в текущей позиции, а 16 — в виде единицы добавляется к следующему по старшинству разряду, в данном случае — к первому. То же правило применяется к остальным разрядам слагаемых:

$$\begin{array}{r} A7_{16} \\ \underline{+1EF_{16}} \\ 296_{16} \end{array}$$

Ответ: 296_{16}.

Примеры для самостоятельного решения

$17E_{16} + E8_{16} = ?_{16}$	$1A_{16} + B4B_{16} = ?_{16}$	$A24_{16} + 2CC_{16} = ?_{16}$
$A7_{16} + E4_{16} = ?_{16}$	$D2_{16} + BF_{16} = ?_{16}$	$CA4_{16} + FF_{16} = ?_{16}$
$25_8 + 73_8 = ?_8$	$45_8 + 63_8 = ?_8$	$44_8 + 26_8 = ?_8$

Преобразования между позиционными системами счисления

Преобразование числа из любой недесятичной системы счисления в десятичную сводится к использованию формулы (1) (см. примеры). Преобразование числа из десятичной в иную недесятичную систему сводится к разложению этого числа по степеням нового основания и выполняется двумя способами:

- для *целой* части числа методом деления с остатком (рис. 1). В качестве делителя используется основание системы счисления, в которую осуществляется преобразование. Частное на каждом шаге деления записывается слева от черты, а из остатков формируется искомое число;

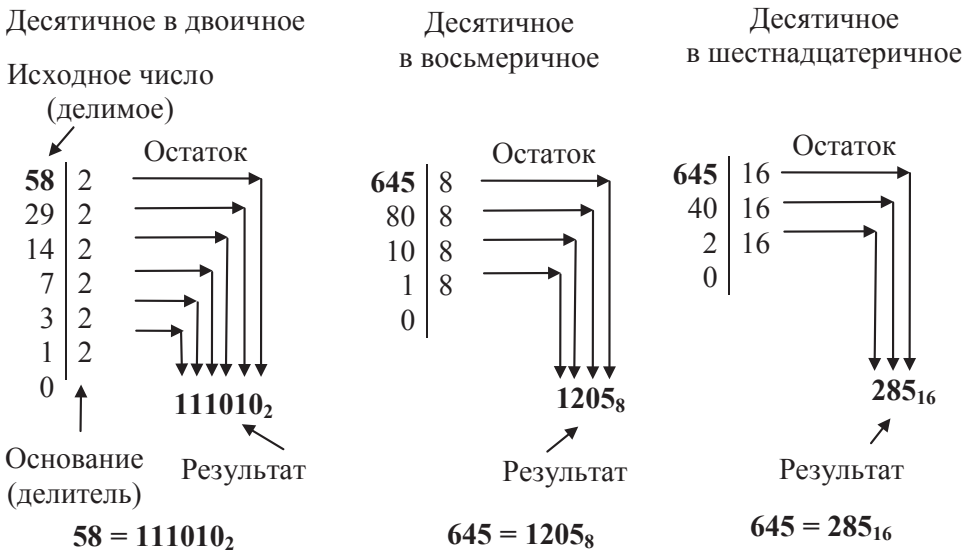


Рис. 1. Метод деления с остатком

- для *дробной* части числа методом последовательного умножения (рис. 2). В данном случае на каждом шаге на основание системы счисления умножается только дробная часть и фиксируется цифра, переходящая в нулевой разряд. Преобразование заканчивается при нуле или периоде в дробной части очередного значения либо по достижении требуемой точности.

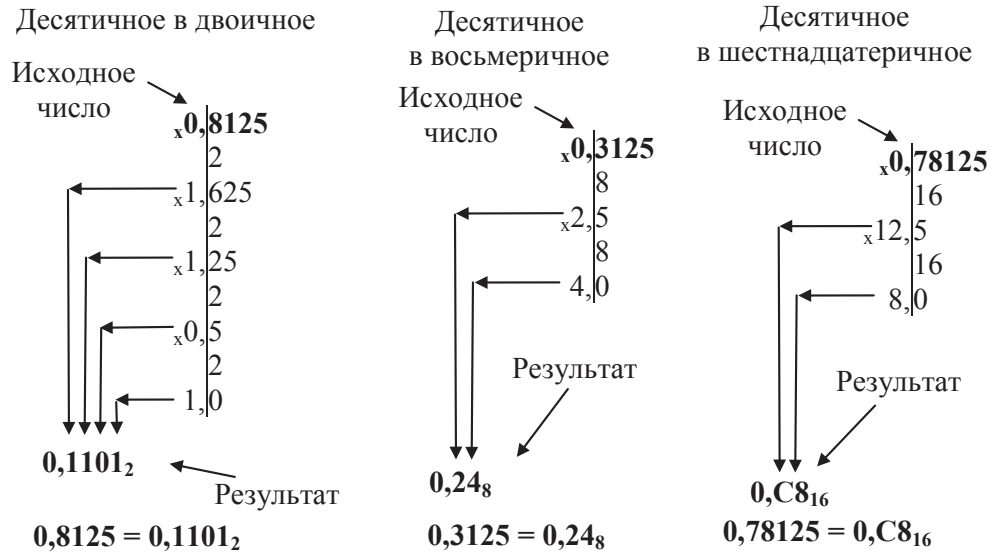


Рис. 2. Метод последовательного умножения

Взаимные преобразования между 2, 8 и 16-й системами основаны на родственности их оснований. Обратите внимание на следующую закономерность в предлагаемой далее таблице соответствия (табл. 3): каждые три разряда двоичной системы соответствуют одной цифре восьмеричной системы; каждые четыре двоичных разряда — одной цифре шестнадцатеричной системы.

Таблица 3

	Система счисления			
	десятичная	двоичная	восьмеричная	шестнадцатеричная
Значение числа	0	0000	0	0
	1	0001	1	1
	2	0010	2	2
	3	0011	3	3
	4	0100	4	4
	5	0101	5	5
	6	0110	6	6
	7	0111	7	7
	8	1000	10	8
	9	1001	11	9
	10	1010	12	A
	11	1011	13	B
	12	1100	14	C
	13	1101	15	D
	14	1110	16	E
	15	1111	17	F

Это позволяет осуществлять преобразования соответственно между двоичной и восьмеричной, а также двоичной и шестнадцатеричной системами простой перекодировкой. Таким же способом можно преобразовать число между восьмеричной и шестнадцатеричной системами, используя в качестве промежуточного результата двоичное значение.

Пример

двоичная $\rightarrow$ восьмеричная: $000111010{,}1101_2 = 72{,}64_8$

двоичная $\rightarrow$ шестнадцатеричная: $000111010{,}001011_2 = 3A{,}2C_{16}$

восьмеричная $\rightarrow$ шестнадцатеричная: $34{,}51_8 = 011100{,}101001_2 = 1C{,}A4_{16}$

Примеры для самостоятельного решения

Выполните преобразования:

$101110{,}01_2 = ?_{16}$	$11011{,}011_2 = ?_{16}$	$11000{,}11011_2 = ?_{16}$	$110{,}1111_2 = ?_{16}$
$10111{,}101_2 = ?_{10}$	$11{,}011_2 = ?_{10}$	$1001{,}111_2 = ?_{10}$	$101{,}0111_2 = ?_{10}$
$123{,}4_8 = ?_{10}$	$133{,}2_8 = ?_{10}$	$117{,}6_8 = ?_{10}$	$22{,}4_8 = ?_{10}$
$142{,}33_8 = ?_{16}$	$37{,}12_8 = ?_{16}$	$13{,}17_8 = ?_{16}$	$254{,}321_8 = ?_{16}$

Выполните действия:

$1D_{16}+32_8+101011010_2=?_{10}$	$12_{16}+25_8+101101110_2=?_8$
$2E_{16}+101010110_2+43_8=?_{16}$	$59_{16}+13_8+101011110_2=?_{16}$
$46+101001110_2+1C_{16}=?_8$	$1D_{16}+16_8+110101110_2=?_{10}$
$1C_{16}+53_8+10101010_2=?_8$	$52_{16}+67_8+10101110_2=?_8$
$28_{16}+1010110_2+43_8=?_{10}$	$34_{16}+146_8+10111110_2=?_{16}$
$61+10100110_2+1C_{16}=?_{16}$	$1D_{16}+102_8+11010110_2=?_{10}$

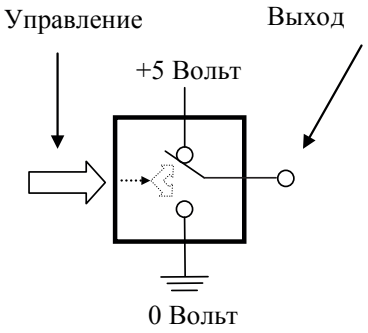
Внутреннее представление числовых данных

В компьютерах первого поколения, например в *Марк-1*, за основу была принята десятичная система. Из-за этого выбора, а также и из-за того, что эта вычислительная машина была реализована на электромеханических элементах, она имела очень большие размеры и чрезвычайно сложную схему (содержала 3304 реле, соединенных проводами общей протяженностью более 800 км). Компьютер *Эниак*, разработанный в 1945 году на основе десятичной системы, также имел до-

статочно солидные размеры: состоял из 18000 электронных ламп, весил 30 тонн, занимал площадь 200 кв. м. и потреблял мощность около 150 кВт.

Компьютер, как и любое другое цифровое устройство, оперирует с так называемыми разрешенными уровнями сигнала. Количество этих уровней соответствует используемой в устройстве системе счисления, а точнее — ее алфавиту. Так, десятичная система требует наличия десяти таких разрешенных уровней для каждой цифры от 0 до 9. В то же время двоичная система обходится всего двумя разрешенными уровнями сигнала — для нуля и единицы. Данное обстоятельство обеспечивает повсеместное использование двоичной системы в современных информационных устройствах. Поскольку в качестве характеристики сигнала в цифровых устройствах чаще всего используется напряжение, принято, что нуль соответствует потенциалу от –0,5 до +0,5 В., а единица от +2 до +5,5 В (так называемая *позитивная пятивольтовая логика*). Следует упомянуть, что первым собрал компьютер на основе двоичной системы немецкий ученый Конрад Цузе в 1941 году. Компьютер получил название «*Z3*». Оригинальным в этом устройстве было еще и то, что команды во время его работы вводились с помощью предварительно отперфорированной кинопленки.

Элементарным устройством для хранения значения двоичного разряда является триггер, или бистабильный элемент. На рис. 3 приведена условная, т. н. *переключательная* или *коммутационная* схема, поясняющая принцип его работы: под действием входного сигнала, обозначенного на рисунке как внешнее управление, триггер может принимать одно из двух состояний и находиться в каждом из них неограниченное время.



Положение переключателя	Физическое значение сигнала	Логическое значение сигнала
Верхнее	+5 Вольт	1
Нижнее	0 Вольт	0

Рис. 3. Логическая (переключательная) схема триггера

На практике логическое значение сигнала определяется диапазоном напряжений. Так, в схемах на основе биполярного транзистора единичное значение соответствует напряжению >= 2 В, а нулевое <= 0,8 В.

Как уже было упомянуто, один разряд двоичной системы счисления одновременно является еще и наименьшей порцией информации, **битом** (*bit* — *binary digit* — двоичная цифра). Восемь двоичных разрядов образуют **байт** (*byte* — слог, рис. 4). Кроме бита и байта, используют производные единицы измерения количества информации:

Килобайт: 1 Кбайт = 2^{10} байт = 1024 байт;

Мегабайт: 1 Мбайт = 2^{10} Кбайт = 2^{20} байт = 1 048 576 байт;

Гигабайт: 1 Гбайт = 2^{10} Мбайт = 2^{30} байт;

Терабайт: 1 Тбайт = 2^{10} Гбайт = 2^{40} байт;

Петабайт: 1 Пбайт = 2^{10} Тбайт = 2^{50} байт.

В одном байте размещаются естественным образом целые двоичные числа в диапазоне от нуля (все биты нулевые) до числа 11111111_2 = = 255 (все разряды единичные). Тот же результат дает формула значения максимального числа в позиционной системе счисления $\max = b^n - 1$, где b — основание системы счисления, n — количество разрядов числа.

Номер разряда (бита)	7	6	5	4	3	2	1	0
Содержимое разряда	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**1**

Рис. 4. Схема байта

Таким образом, наибольшее число, которое можно разместить в байте, $\max = 2^n - 1 = 2^8 - 1 = 255$. В качестве вывода: в байте можно представить любое целое положительное число из диапазона [0, ..., 255] — всего **256** значений.

Два смежных байта иногда называют словом (рис. 5).

Старший байт								Младший байт							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	1	1	0	0	0	1	1	1	1	0	0	1

Рис. 5. Схема слова

Рассуждая аналогичным образом, несложно установить, что в слове может быть представлено любое число из диапазона [0,.., 65 535], всего **65536** значений. Если выполнить подобную оценку для 32-раз-

рядного и 64-разрядного двоичного числа, получим диапазоны форматов, использующихся в настоящее время для представления *положительных* (*беззнаковых*) целых чисел (табл. 4).

Таблица 4

Целый беззнаковый формат	Количество разрядов	Диапазон значений
Байтовый	8	0 ... 255
Короткий	16	0 ... 65 535
Средний	32	0 ... 4 294 967 295
Длинный	64	0 ... 18 446 744 073 709 551 615

В общем случае формат — это набор правил, используемых для представления элемента соответствующего типа данных. Основной характеристикой формата является его размер. Кроме этого, форматом определяются количество, назначение, порядок следования и размеры составляющих его полей.

Для целочисленных данных различают беззнаковые и знаковые форматы, размером один, два, четыре и восемь байт. Беззнаковые форматы состоят из единственного поля.

Примеры для самостоятельного решения

Определите, какое наибольшее положительное число можно разместить

- в четырех двоичных разрядах?
- в шести битах?
- в трех разрядах восьмеричной системы?
- в двух разрядах шестнадцатеричной системы?
- в трех разрядах шестнадцатеричной системы?
- в десяти разрядах двоичной системы?
- в двадцати битах?

Предшествующее изложение данной главы относилось к беззнаковой форме представления, так как речь шла о положительных целых числах, все разряды которых рассматриваются как значимые. В знаковой форме представления числовых данных соответственно добавляется новый вид информации — знак числа. По этой причине все знаковые целочисленные форматы состоят из двух полей: в старшем разряде кодируется знак (0 — число положительное или нуль, 1 — от-

рицательное), а остальные разряды определяют значение (модуль) числа (рис. 6). При этом уменьшается значение максимального числа, представимого в заданном количестве разрядов, но ширина диапазона остается прежней (табл. 5).

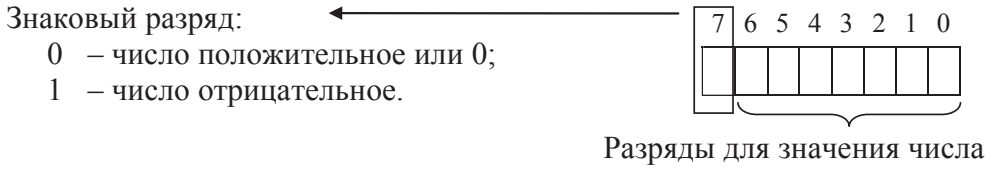


Рис. 6. Байтовый знаковый формат

Таблица 5

Целый знаковый формат	Число разрядов	Диапазон значений	Ширина диапазона
Байтовый	8	–128 ... 127	256
Короткий	16	–32 768 ... 32 767	65536
Средний	32	–2 147 483 648 ... 2 147 483 647	4 294 967 296
Длинный	64	–9 223 372 036 854 775 808 ... 9 223 372 036 854 775 807	2^{64}

Приведенные выше рассуждения о представлении отрицательных целых чисел касались так называемой формы значение со знаком (ФЗСЗ). Данная форма обладает достаточно высокой наглядностью, но не может быть использована для непосредственного выполнения арифметических операций с отрицательными числами.

Отрицательные числа в цифровых системах представляются в форме дополнения до двух, речь о которой пойдет в следующей главе.

Для представления действительных чисел, т. е. чисел, имеющих дробную часть (иначе их называют числа с плавающей точкой в отличие от целых чисел — с фиксированной точкой), разработаны специальные форматы.

Вещественный формат одинарной точности или короткий (рис. 7) обеспечивает точность шесть — семь значащих цифр после запятой (в десятичном представлении), размещается в 32 битах, часто называется *float* и включает в себя три поля:

- знаковый разряд (1 бит);
- смещенный порядок (8 бит), определяемый по формуле $p_{смещенный} = p_{реальный} + 2^7 - 1$ (всегда целый и положительный);

- мантисса — нормализованная двоичная дробь (23 бита). Нормализация заключается в подборе такого порядка, чтобы в целой части двоичного числа всегда находилась одна единица. Эта единица не несет информационной нагрузки, в формате не хранится, но должна учитываться в операциях с вещественным числом.

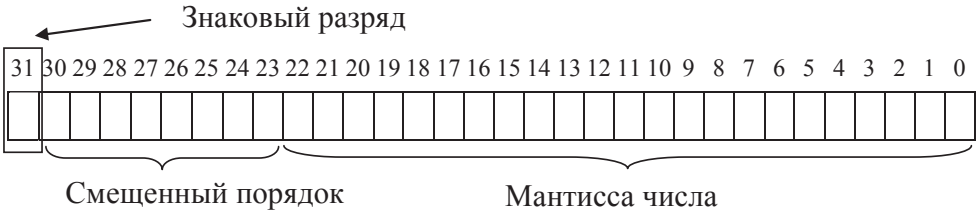


Рис. 7. Короткий вещественный формат

Пример. Установите, какому десятичному числу соответствует значение 4353 $E000_{16}$, заданное в коротком вещественном формате.

Решение. В двоичной системе исходное значение имеет вид

$0100\,0011\,0101\,0011\,1110\,0000\,0000\,0000_2$,

поэтому знаковый бит = 0_2, следовательно, число положительное;

смещенный порядок = $1000\,0110_2$ = 134;

реальный порядок =134–127 = 7;

нормализованная мантисса = $1{,}10100111110_2$.

Ответ. Искомое число = $1{,}1010\,0111\,110_2 2^7$ = $1101\,0011{,}111_2$ = D3, E_{16} = 211,875.

Примеры для самостоятельного решения

Определите числа, представленные в вещественном формате одинарной точности значениями: а) 42825000_{16}; б) $C29CE000_{16}$.

Преобразуйте в вещественный формат одинарной точности следующие числа: в) 120,6875; г) –62,65625.

Вещественный формат удвоенной точности (или *double*) имеет ту же структуру, что и *float*, но имеет размер уже 8 байт (64 бита), которые между полями делятся следующим образом:

1 разряд — знак;

11 разрядов — смещенный порядок;

52 разряда — нормализованная матрица.

Соответственно смещение порядка составляет в этом случае 1023 ($2^{10}-1$). Обеспечивает данный формат точность до 15–16 знаков в десятичном представлении числа.

С появлением в составе процессоров специализированных 128-разрядных регистров появилась аппаратная возможность для реализации вещественного формата повышенной точности, который получил название *decimal*. Данный формат имеет размер 16 байт, состоит из 5 полей и его структура соответствует следующей схеме:

127	126 120	119 112	111 96	95 0
S	Нули (7 бит)	Множитель (8 бит)	Нули (16 бит)	Целое число (96 бит)

Здесь: *S* — знаковый бит, Множитель (или масштабирующий множитель — ММ) — это целое число, которое может принимать значения от 0 до 28. Математически ММ представляет собой отрицательную степень в экспоненциальном представлении вещественного числа при его нормализации к нулевой дробной части (по сути, ММ — это количество цифр в дробной части).

Пример. Преобразуйте в вещественный формат повышенной точности число 1234,456.

Решение. $1234{,}456 = 1234456\,10^3 = 12D618h\ 10^3$ (ММ = 3).

Ответ. Внутреннее представление искомого числа в вещественном формате повышенной точности

$00\,03\,00\,00\,00\,00\,00\,00\,00\,00\,00\,00\,00\,00\,12\ D6\,18_{16}$.

Примеры для самостоятельного решения

Определите числа, представленные в вещественном формате повышенной точности следующими значениями:

а) 00 07 00 00 00 00 00 00 00 00 00 00 03 3d 0b 33 h;

б) 00 04 00 00 00 00 00 00 00 00 00 00 00 00 d4 31 h.

Преобразуйте в вещественный формат повышенной точности следующие числа: в) 123,456789; г) –98,7654321.

Текстовые данные

Сравнительно быстро персональный компьютер стал незаменим для подготовки или обработки текстов. Любой текст это ничто иное, как набор букв, цифр, знаков препинания, других спецсимволов. Компьютер как цифровое устройство имеет возможность обрабатывать данные так же только в цифровом виде. Поэтому для представления текстовых данных каждому символу ставится в соответствие цифровой код. Существует множество различных систем кодирования. Так, наиболее распространенной для компьютеров, работавших под управлением операционной системы MS–DOS, была система ASCII (*American Standart Code for Information Interchange* — американский стандартный код для обмена информацией). Этот стандарт определял код размером один байт и состоял из двух частей. Первая часть, кодирование в которой осуществляется в 7 битах байта, называлась основной таблицей ASCII и содержала 128 различных кодов от 0 до 127. При этом восьмой, старший бит, был первоначально зарезервирован для контроля четности (система достоверизации данных).

В итоге, в первой части таблицы ASCII были представлены:

- управляющие символы, такие как возврат каретки ($CR = 13$), перевод строки ($LF = 10$) и другие (коды с 0 по 31);
- цифры десятичной системы с 0 по 9 (коды с 48 по 57);
- знаки препинания и спецсимволы (коды 32–47, 58–64, 91–96, 123–126);
- большие (коды 65–90) и маленькие (коды 97–122) буквы латинского алфавита.

Интересы распространений компьютеров за пределами англоговорящих стран привели к необходимости отказаться от размещения в коде символа бита контроля четности, а использовать старший разряд как значащий. Это позволило сформировать вторую часть таблицы ASCII, в которой используются для размещения кода уже все 8 бит байта. Обычно эту часть называют расширением таблицы ASCII, она содержит 128 кодов в диапазоне от 128 по 255 различных для национальных алфавитов. Дополнительно во второй части таблицы ASCII были закодированы символы псевдографики, математические символы и некоторые другие. В нашей стране была распространена таблица под названием кодовая страница 866 (CP866 — *Code Page866*), содержащая основную таблицу ASCII и вариант второй части табли-

цы с русским алфавитом в так называемой *альтернативной кодировке*.

В операционной системе Windows, локализованной для использования в русскоговорящих странах, использовалась таблица кодировки под названием кодовая страница 1251 (CP1251). Строчные, прописные буквы латинского алфавита, цифры и обычные знаки препинания в ней были закодированы так же, как и в CP866. По-разному представлялись спецсимволы и буквы русского алфавита. По этой причине русские тексты, подготовленные, например в системе *MS–DOS*, перед их использованием в системе *Windows*, должны были быть перекодированы (конвертированы).

Имеется еще одна восьмиразрядная кодировка текстовой информации, известная под названием КОИ–8, которая первоначально была стандартом в странах бывшего СЭВ.

На смену системам кодирования с восьмибитовым кодом пришла 16-битная система Unicode, которая потенциально может содержать до 65536 различных символов. Такой диапазон вполне достаточен для размещения в одной таблице символов для всех основных языков планеты. В частности, кириллические коды занимают в ней диапазон от 400_{16} до $52F_{16}$.

Графические данные

Имеется несколько систем кодирования графических данных:

а) в случае, когда графическое изображение представляется состоящим из набора точек, говорят о *растровой графике*;

б) если же базовым элементом изображения является линия или дуга, описываемые математически как отдельные объекты, то это *векторная графика*.

Рассмотрим более подробно принципы кодирования изображения в растровой графике. Состояние каждой точки растра кодируется набором бит (двоичных разрядов) соответствующей длины. Длина определяется по формуле

$$n = \log_2 Color,$$

где n — количество необходимых бит, $Color$ — число состояний точки или цветность изображения.

В предельно простом случае (светится — не светится *Color* = 2) для информации о состоянии точки достаточно одного бита. Для улучшенных чернобелых изображений используют 256 градаций серого цвета и поэтому состояние каждой точки растра определяется байтом данных.

Для кодирования цветных изображений используется одна из следующих моделей:

- **Аддитивная RGB-логика**. Данная модель используется при электронном воспроизведении (на дисплее: восприятие прямых лучей). Любой цвет здесь представляет собой суммирование в различной пропорции трех основных цветов: **Red** (красный), **Green** (зеленый), **Blue** (синий). Если состояние каждого основного цвета описывается одним битом, отвечающим только за его присутствие в суммарном, получается восемь цветов и каждая точка растра кодируется тремя битами ($\log_2 8 = 3$). В случае 16-цветного режима необходимы четыре бита. Иметь 256 цветовых оттенков каждой точке растра позволяет байт данных. Все вышеперечисленные цветные режимы называются индексными, поскольку на самом деле значение двоичного числа, отвечающего за состояние точки, используется в качестве номера строки в таблице, называемой палитрой. Значение строки таблицы определяет интенсивность каждого из основных цветов. Режим **High Color** предусматривает кодирование состояния каждой точки растра с помощью 16-разрядного двоичного числа и непосредственно управляет системой цветовоспроизведения дисплея. С учетом особенностей зрения человека эти разряды между основными цветами делятся в следующей пропорции: **5:6:5.** В режиме **True Color** информация о состоянии точки кодируется в 24-х разрядах, которые между основными цветами делятся поровну.
- **Субстрактивная CMYK-логика**. Компонентами здесь служат цвета, полученные вычитанием основных цветов из белого:
  Cyan (фиолетовый) = белый — красный;
  Magenta (пурпурный) = белый — зеленый;
  Yellow (желтый) = белый — синий, к которым добавляется еще чисто черный цвет (Blac**K**).

Модель CMYK используется для вывода изображений на бумажный носитель, когда цветовосприятие обеспечивают отраженные лучи. Для кодирования состояния каждой точки используется поле в 32 разряда, которое между основными цветами делится поровну, по одному байту.

Арифметика дополнений до двух

Основной объем работы в любой цифровой системе выполняет уже упомянутое арифметико-логическое устройство. Центральным элементом последнего является сумматор — схема, выполняющая двоичное сложение. Как правило, остальные три базовые арифметические операции (вычитание, умножение и деление) выполняются также сумматором. При этом, чтобы научить сумматор умножению, достаточно реализовать сдвиг и свести умножение к серии последовательных сложений со сдвигом. Несколько сложнее обстоит дело с вычитанием. Чтобы сумматор адекватно воспринимал отрицательные числа, они должны быть представлены в форме дополнений до двух (ФДД). Дополнение до двух — это эквивалент, использование которого в операции суммирования (вместо заданного числа) позволяет получить верный результат.

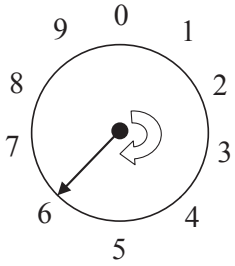


Рис. 8. Механический одноразрядный сумматор

Суть метода поясним на примере из десятичной системы. Представим некоторый одноразрядный механический сумматор в виде циферблата (рис. 8), на котором расположены цифры данной системы, и указателя на подвижной оси, способного вращаться только в направлении часовой стрелки. С помощью этого прибора, прежде всего, выполним сложения, например, 6 + 2. Первоначально стрелка — указатель сумматора устанавливается на цифру 6 (первое слагаемое), затем передвигается на два деления вперед (количество делений соответствует второму слагаемому) и тем самым оказывается на цифре 8 — результате операции.

Далее на этом же устройстве выполним операцию вычитания: 6–2, не забывая, что стрелка в противоположном направлении перемещаться не может. Задача выполнима при замене числа –2 в операции сложения его дополнением — числом 8:

$$6–2 = 6 + (–2) = 6 + \text{Доп}_{10}\,(–2) = 6 + 8 = 4,$$

где $\text{Доп}_{10}\,(–2) = 10–2 = 8$ — дополнение числа –2 до десяти (основание системы счисления).

Отметим, что дополнение до 10 числа –2 (т. е. восемь) можно также получить альтернативным образом:

- найти дополнение исходного числа до 9 (т. е. семь);
- прибавить к полученному дополнению единицу.

Несмотря на кажущуюся искусственность, для двоичной системы такой путь, более простой и быстрый, называется логическим способом получения дополнения до двух:

1. Исходное отрицательное число подготавливается в форме значения со знаком;

2. Затем выполняются три следующих шага:

- шаг 1 — инвертировать все биты числа, кроме знакового, получив таким образом дополнение до единицы;
- шаг 2 — прибавить единицу к полученному числу, получив дополнение до двух;
- шаг 3 — восстановить единицу в знаковом разряде.

Нижеследующие примеры данной главы используют байтовый формат.

Пример. Получить для числа –5 дополнение до двух.

Прежде всего, запишем число –5 в форме значения со знаком:

$$-5 = 10000101 \text{ (ФЗСЗ)},$$

затем выполним три шага получения дополнения:

шаг 1 1111010

шаг 2 +_____1

1111011

шаг 3 11111011 – результат в ФДД.

Если к полученному результату еще раз применить процедуру получения дополнения, получится первоначальное число. Таким образом, доказывается цикличность дополнений.

Действия, имитирующие работу сумматора с отрицательными числами, можно обобщить в виде алгоритма вычитания.

- Этап 1: преобразовать отрицательное число (числа) из формы «значения со знаком» в форму «дополнение до двух».
- Этап 2: выполнить операции двоичного сложение над всеми битами, включая знаковые. В случае, если имеет место перенос из самого старшего разряда, можно разместить единицу в допол-

нительном разряде. У сумматора данный разряд реализован отдельным триггером и называется флагом переноса (*C* — *carry*).

- Этап 3: если знаковый бит суммы равен единице, то результат — отрицательное число в форме дополнения до двух. Для получения числа в форме «значение со знаком» необходимо от результата вновь получить дополнение до двух (используя свойство цикличности дополнений).

Пример. $6_{10}-15_{10} = ?$

Этап 1: $-15 = 10001111_2$ (ФССЗ)

шаг 1 $\quad 1110000_2$ (инвертирование)

шаг 2 $\quad +\underline{\qquad 1}$

шаг 3 $\quad 11110001_2$ дополнение до двух числа -15.

Этап 2: число $6 = 00000110_2$

$$\begin{array}{l} \;\;00000110_2 \\ \underline{+\;11110001_2} \text{ (число } -15 \text{ (ФДД))} \end{array}$$

11110111_2 — результат в ФДД, т. к. отрицательный.

Этап 3:

$$\begin{array}{l} \;\;0001000_2 \\ \underline{+\qquad\quad 1} \end{array}$$

$10001001_2 = -9$ — окончательный результат.

Пример. $-6-4 = ?$

$-6 = 10000110_2 \quad -4 = 10000100_2$.

Этап 1: $\quad 1111001_2 \qquad\qquad 1111011_2$

$$\begin{array}{r} \underline{+\qquad\quad 1} \\ 11111010_2 \end{array} \qquad\qquad \begin{array}{r} \underline{+\qquad\quad 1} \\ 11111100_2 \end{array}$$

Этап 2: $\quad 11111010_2$

$$\begin{array}{r} +\;\underline{11111100_2} \\ 11110110_2 \end{array}$$

Этап 3: $\quad 0001001_2$

$$\begin{array}{r} +\underline{\qquad\quad 1} \\ 10001010_2 \end{array}$$

$= -10$ — результат.

Ситуация переполнения

При выполнении операции двоичного сложения можно не учитывать перенос из старшего разряда и при этом получать правильные результаты. Однако так бывает не всегда. Покажем это на примерах.

Пример. 65 + 65 =?

```
            01000001
Этап 2: + 01000001
            10000000
Этап 3:     1111101
          +        1
           11111110 = –126 — результат.
```

Пример. –127–125 =?

Этап 1:

Доп$_2$ (–127) = 10000001

Доп$_2$ (–125) = 10000011

Этап 2: 00000100

— результат.

Результаты в обоих примерах ошибочные, поскольку оказались за пределами диапазона возможных значений (для используемого здесь целочисленного байтового формата он составляет, как уже было упомянуто, от –128 до +127). В подобных случаях сумматор должен распознавать ошибочную ситуацию и либо остановить вычисления, либо просигнализировать об особой ситуации. Несложно заметить, что предпосылкой переполнения является одинаковый знак слагаемых и отличный от него — знак результата.

Ситуация переполнения фиксируется состоянием флага переполнения. Для удобства триггеры, используемые в управлении вычислительным процессом, группируются вместе, образуя специальное устройство процессора, называемое флаговым регистром (*F* — *flag*).

Алгоритм идентификации ситуации переполнения представлен на рис. 9.

Таким образом, для выполнения операций сложения и вычитания арифметико-логическое устройство (сумматор) должно быть в состоянии выполнять проверку на знак, инвертирование и поразрядное сложение с переносом. Как уже упоминалось, если к этому набору добавить операции сдвига, то с помощью сумматора окажется возможным дополнительно выполнять операции умножения и деления двоичных чисел.

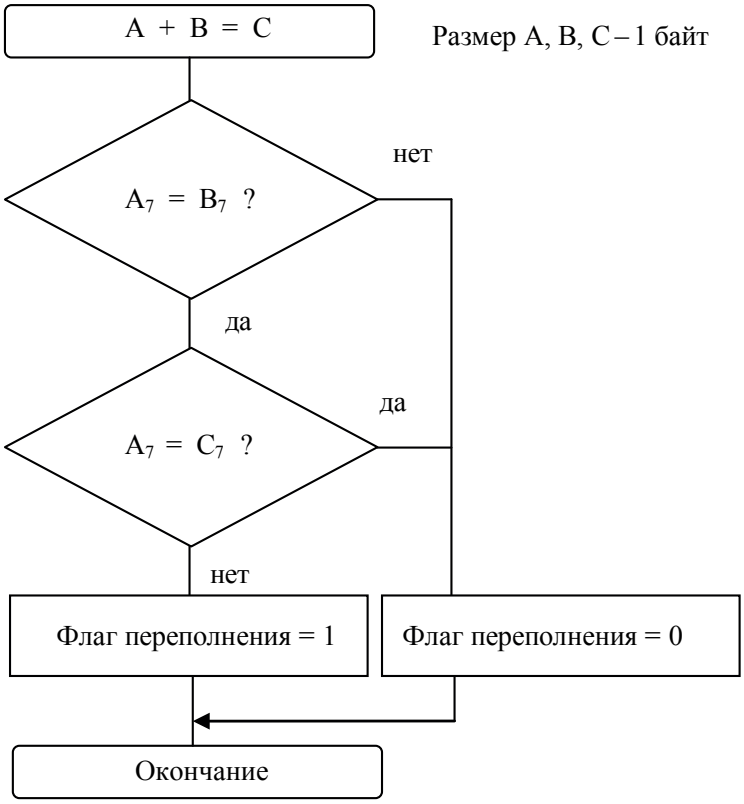


Рис. 9. Алгоритм идентификации переполнения

2. Аппаратные основы обработки данных

Любая современная информационная система представляет собой объединение достаточно большого количества устройств, главным из которых является процессор. Именно способность процессора выполнять под управлением соответствующей программы обработку тех либо иных данных обеспечивает подобным системам предельно универсальный характер. Такие системы принято относить к системам с *гибкой логикой*, в отличие от систем с *жесткой логикой*, функциональность которых определяется их конструкцией и не может быть изменена в процессе эксплуатации.

В составе процессора можно выделить следующие основные узлы и элементы (рис. 10):

- регистры — специальные устройства, предназначенные для подготовки данных перед выполнением команды процессором и для записи результата операции. Регистр можно рассматривать как последовательно размещенные триггеры. Как правило, отдельная группа регистров предназначена для размещения адресов;
- арифметико-логическое устройство (АЛУ) — схема, непосредственно выполняющая обработку данных. Основной операцией АЛУ является суммирование двоичных разрядов (отсюда и второе название АЛУ — «сумматор»);
- дешифратор кода операции;
- схема управления и синхронизации.

Основное назначение процессора — выполнение операций над данными, образующими программу. Для того чтобы процессор под управлением последовательности команд был способен обрабатывать данные в составе компьютера или иной информационной системы, в общем случае, в дополнение к процессору должны быть устройства RAM- и ROM-памяти, порты ввода/вывода, контроллеры прерыва-

ний и еще ряд специализированных устройств. Большая часть из перечисленного — полупроводниковые схемы. Необходимые сведения по их конструкции и принципам функционирования содержит материал следующих разделов.

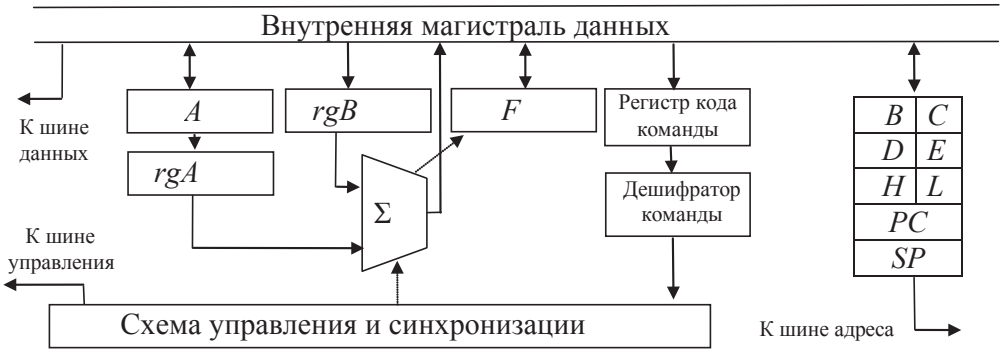


Рис. 10. Основные подсистемы процессора

Краткое введение в цифровую электронику

Примесные полупроводники

Основой полупроводниковой техники является кремний — элемент IV группы таблицы Менделеева, который относится к природным полупроводникам. Полупроводники — это класс веществ, занимающих промежуточное место между проводниками и диэлектриками. Проводимость природных полупроводников обусловлена наличием в них одновременно свободных носителей заряда различных знаков. Электроны (носители отрицательного заряда) появляются вследствие так называемого теплового заброса части электронов из валентной зоны в зону проводимости, а дырки (носители положительного заряда) представляют собой вакантное место в валентной связи, которое электрон как раз и покинул. Природные носители заряда принято называть неосновными, так как основные носители в полупроводнике получают путем легирования. Для этого в химически чистый расплавленный кремний добавляют элементы III или V группы (рис. 11).

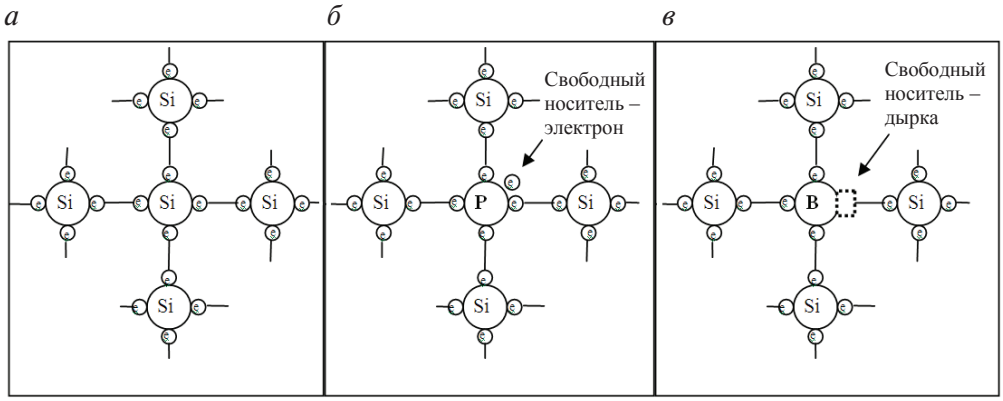


Рис. 11. Получение примесного полупроводника:
а — природный полупроводник; *б* — полупроводник *n*-типа; *в* — полупроводник *p*-типа

На рис. 11, *а* показан фрагмент кристаллической решетки кремния, в которой каждый атом за счет четырех валентных электронов образует соответственно четыре валентные связи с такими же соседними атомами. Если в кремний добавляют, например фосфор, принадлежащий V группе и имеющий 5 валентных электронов, то после замены в узле кристаллической решетки атома кремния примесным атомом (рис. 11, *б*) один (пятый) электрон фосфора не может образовать валентную пару и остается несвязанным, обеспечивая проводимость полупроводника *n*-типа. В случае использования бора, элемента третьей группы, в одной из валентных пар внедренного атома появляется вакантное не занятое электроном место, обеспечивая полупроводнику проводимость *p*-типа. После легирования концентрации примесных носителей заряда превосходят концентрации природных носителей до сотни тысяч раз (рис. 11, *в*).

Существует два способа управления проводимостью полученных таким образом полупроводников:

- управление электрическим током. В этом случае проводимость обеспечивают, как правило, носители обоих знаков, а объектом управления является биполярный транзистор;
- управление электрическим полем. Здесь проводимость обеспечивают носители только одного знака, а транзистор называется униполярным, или полевым.

Биполярный транзистор и схемы на его основе

Работа биполярного транзистора во многом определяется свойствами неуправляемого *pn*-перехода, который входит в его состав. В свою очередь, *pn*-переход представляет собой устройство из двух смежных разнопроводящих объемов полупроводника (рис. 12, *а*). За счет большой разности концентраций носителей заряда через границу между этими объемами возникает диффузионный поток электронов из области *n*-типа в область *p*-типа. За счет этого в приграничных областях образуются обедненные основными носителями слои, имеющие большое удельное сопротивление и объемные заряды разного знака. Последние обеспечивают граничную напряженность, препятствующую протеканию тока при прямом включении *pn*-перехода (рис. 12, *а*) при небольших значениях приложенного напряжения. Ток появляется и резко увеличивается при достижении напряжением величины примерно 0,6 В (рис. 12, *б*). В данном режиме *pn*-переход открыт и имеет практически нулевое сопротивление.

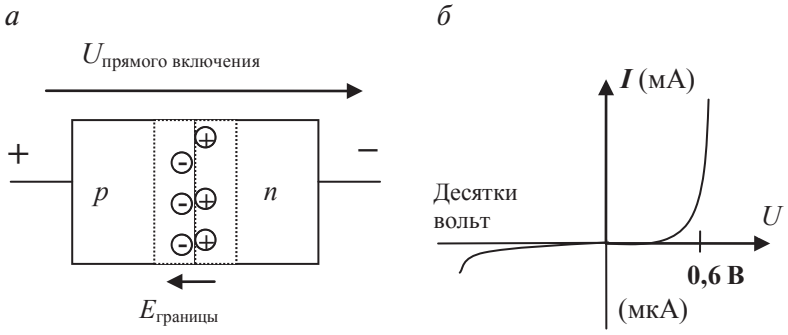


Рис. 12. *pn*-переход:
а — структура *pn*-перехода; *б* — вольтамперная характеристика

Конструктивно биполярный транзистор представляет собой три смежные области полупроводников разной проводимости. Большее распространение в цифровой электронике получили *npn*-транзисторы, в составе которых соответственно две области *n*-типа и одна область *p*-типа (рис. 13, *а*). Напряжение $U_{кэ}$, приложенное между коллектором и эмиттером, — это напряжение питания. Управляющее напряжение $U_{бэ}$ — это напряжение, приложенное соответственно между базой и эмиттером. До тех пор, пока $U_{бэ} <= 0{,}6$ В и переход база–эмиттер закрыт, транзистор также находится в закрытом состоянии и $R_{кэ} = \infty$. Причина этого в том,

что даже при наличии ускоряющего напряжения $U_{кэ}$ невозможен перенос заряда между коллектором и эмиттером через несколько областей разного типа проводимости. Когда $U_{бэ}$ становится > 0,6 В, база-эмиттерный *pn*-переход открывается и электроны из *n*-области эмиттера попадают в *p*-область базы. Часть из них рекомбинируют, а остальные, продолжая движение под действием напряжения $U_{кэ}$, попадают в область коллектора, обеспечивая тем самым ток коллектора и близкое нулю сопротивление между коллектором и эмиттером.

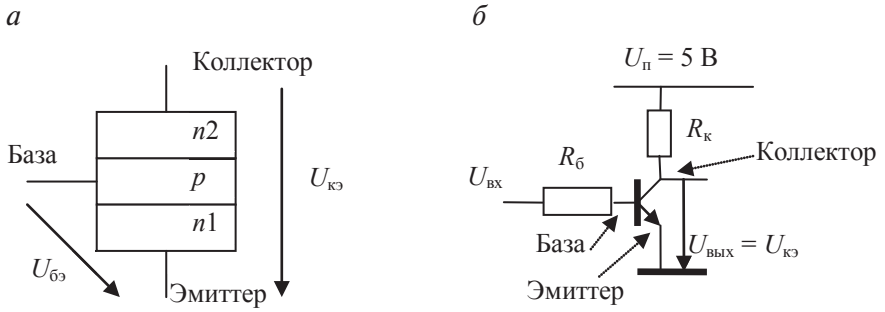
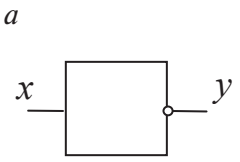


Рис. 13. Биполярный *npn*-транзистор:
а — структурная схема; *б* — биполярный ключ

Схема, приведенная на рис. 13, *б*, представляет собой самое простое цифровое устройство — инвертор, или элемент НЕ (рис. 14). Действительно, при $U_{вх}$ близком 0 В (логический ноль), напряжения на базе недостаточно, чтобы открыть переход база–эмиттер, и транзистор остается закрытым. При этом сопротивление $R_{кэ}$, практически, равно бесконечности (по сути — разрыв цепи) и $U_{вых}$ близко напряжению питания 5 В (логическая единица). Если же $U_{вх}$ близко 5 В, на базе транзистора после падения на $R_б$ остается достаточное напряжение, чтобы открыть транзистор и обеспечить близкое нулю сопротивление $R_{кэ}$ (по сути — закоротку). В этом случае на выходе схемы формируется напряжение близкое 0 В — потенциалу земли (рис. 14).



б

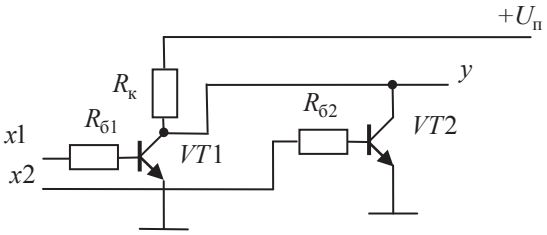
x	*y*
0	1
1	0

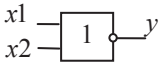
Рис. 14. Элемент НЕ:
а — условное изображение; *б* — таблица состояний

На рис. 14, *а* представлено условное изображение инвертора, а на 16, *б* таблица состояний, описывающая в логических уровнях соответствие входных и выходных сигналов для различных режимов работы.

Не многим сложнее так называемые базисные устройства: элементы ИЛИ-НЕ (рис. 15) и И-НЕ (рис. 16). Базисными их называют потому, что, используя любое из них, можно собрать цифровую схему заданного назначения. Эти элементы получили специальные названия. Так, элемент И-НЕ называют штрихом (или базисом) Шеффера, а элемент ИЛИ-НЕ — стрелкой (или базисом) Пирса.

а *б* *в*



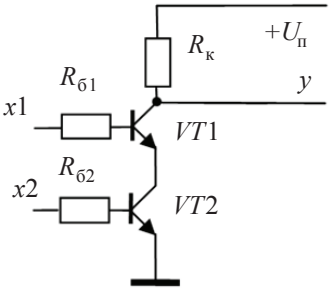


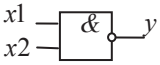
x1	x2	y
0	0	1
0	1	0
1	0	0
1	1	0

Рис. 15. Элемент ИЛИ-НЕ:

а — упрощенная схема; *б* — условное изображение; *в* — таблица состояний

а *б* *в*



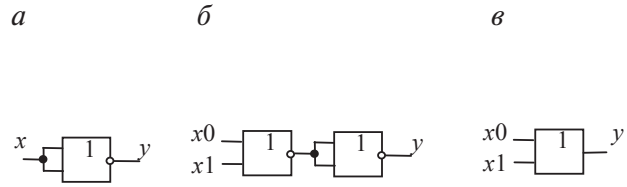


x1	x2	y
0	0	1
0	1	1
1	0	1
1	1	0

Рис. 16. Элемент И-НЕ:

а — упрощенная схема; *б* — условное изображение; *в* — таблица состояний

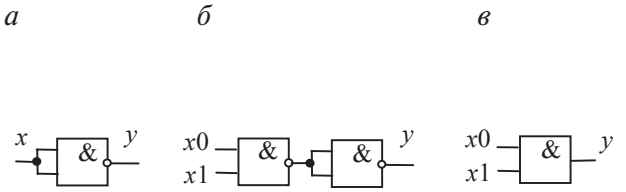
Для получения простых логических функций И, ИЛИ (без выходной инверсии) также используются базисные элементы (рис. 17, 18).



x1	x0	y
0	0	0
0	1	1
1	0	1
1	1	1

Рис. 17. Получение элемента ИЛИ на базе элемента ИЛИ-НЕ:

а — элемент НЕ; *б* — элемент ИЛИ; *в* — условное обозначение; *г* — таблица состояний



x1	x0	y
0	0	0
0	1	0
1	0	0
1	1	1

Рис. 18. Получение элемента И на базе элемента И-НЕ

а — элемент НЕ; *б* — элемент И; *в* — условное обозначение; *г* — таблица состояний

Полевой транзистор и схемы на его основе

Из полевых транзисторов в цифровой электронике наибольшее распространение получили нормально закрытые транзисторы с изолированным затвором. Упрощенная конструкция для варианта *p*-канального транзистора приведена на рис. 19. Полевые транзисторы подобного типа в соответствии с их конструкцией получили аббревиатуру МОП (металл — оксид — полупроводник). По сути, в этом обозначении проговаривается его конструкция: над областью полупроводника (подложка) выполняется слой изолятора — оксида кремния, по которому напыляется тонкий слой проводника (затвор).

При отсутствующем или положительном напряжении на затворе транзистор находится в закрытом состоянии, потому что между *p*-проводящим истоком и *p*-проводящим стоком расположена *n*-проводящая подложка (рис. 19, *а*). Когда на затвор подают отрицательный потенциал, напряженность его электрического поля обеспечивает направленный дрейф неосновных носителей заряда (в данном случае — дырок) из глубин подложки на границу полупроводник–изолятор. Сформированный таким образом проводящий слой (канал *p*-типа) обеспечивает электрическую проводимость между истоком и стоком (рис. 19, *б*).

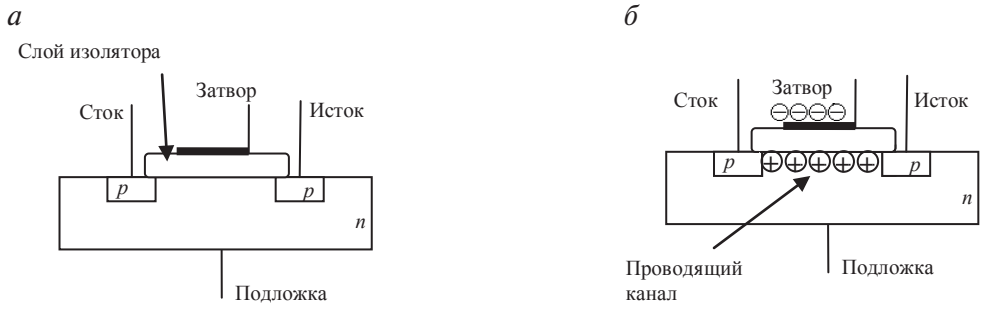


Рис. 19. Конструкция полевого транзистора:

а — конструкция *p*-канального транзистора; *б* — транзистор в открытом состоянии

В конструкции *n*-канального МОП-транзистора все наоборот: подложка *p*-типа, а области истока и стока *n*-типа; открывается транзистор положительным потенциалом затвора. На рис. 20 приведены схемные обозначения полевых транзисторов.

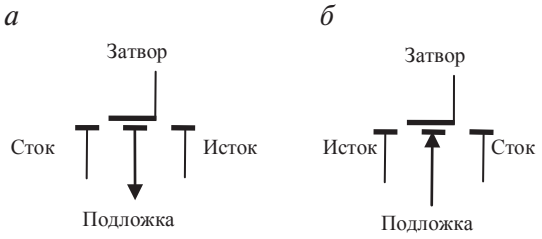


Рис. 20. Схемные обозначения полевых транзисторов:

а — схемное обозначение *p*-канального транзистора; *б* — схемное обозначение *n*-канального транзистора

Как и в случае биполярных транзисторов, наиболее простой цифровой схемой на полевых транзисторах является инвертор (рис. 21).

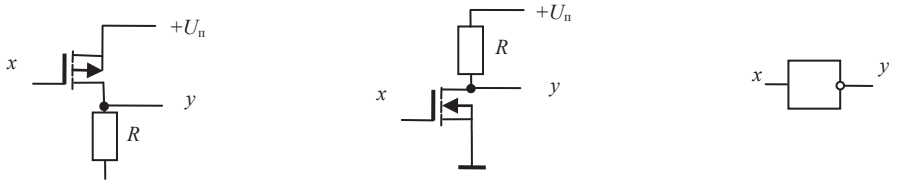


Рис. 21. Инвертор на полевых транзисторах:

а — инвертор на базе *p*-МОП; *б* — инвертор на базе *n*-МОП; *в* — условное обозначение

Для полевых транзисторов приняты несколько иные соглашения по уровням цифрового сигнала: логической единице соответствует напряжение, равное или выше 80 % напряжения питания схемы, а нулю — напряжение от 0 до 0,5 вольт.

Элементы и узлы цифровых систем

Дешифратор

Одним из самых массовых цифровым устройством является дешифратор. В составе процессора он распознает, какой должна быть очередная операция, и называется дешифратором команд, в устройствах памяти он определяет элемент, к которому последует обращение, и называется дешифратором адреса. Принцип работы дешифратора поясняет рис. 22.

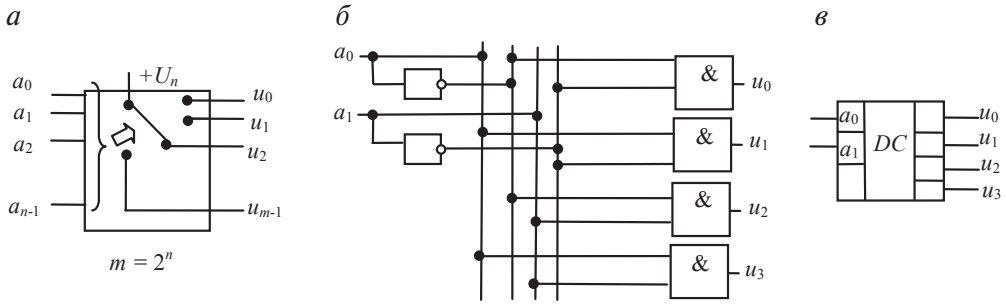


Рис. 22. Дешифратор:

а — переключательная схема; *б* — схема в детализации до логических элементов; *в* — условное изображение

В соответствии со значением, полученным на *n*-входах, дешифратор выделяет единичным (высоким) уровнем сигнала одну из *m*-выходных линий (рис. 22, *а*). По сути, дешифратор выбирает выходную линию, номер которой соответствует входным сигналам, если их рассматривать, как одно двоичное *n*-разрядное число. На рис. 22, *б* приведена схема дешифратора на 2 входа, на рис. 22, *б* — его условное изображение, а табл. 6 содержит описание входных и выходных сигналов. Действительно, при нулевых значениях на входах дешифратор формирует единицу только на линии u_0, а например, при единичных — на линии u_3.

Таблица 6

Таблица состояний дешифратора на два входа

a_1	a_0	u_3	u_2	u_1	u_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Сумматор

Как уже упоминалось выше, основной арифметической операцией в компьютерных системах является сложение. Рассмотрим схему простейшего одноразрядного полусумматора, который способен выполнять сложение двух младших разрядов слагаемых (рис 23, *а*).

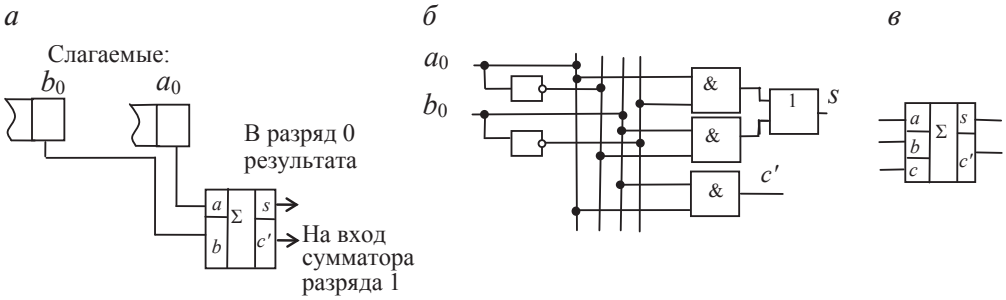


Рис. 23. Полусумматор и сумматор:

а — суммирование младших разрядов полусумматором; *б* — схема полусумматора; *в* — условное обозначение сумматора

Верхняя часть схемы полусумматора формирует нулевой разряд *S* суммы (рис. 23, *б*). В цифровой электронике эта функция получила название *исключающее ИЛИ* — в том смысле, что из единичных результатов логики ИЛИ исключен случай совпадения двух единиц (табл. 7, столбец s). Нижняя часть схемы (рис. 23, *б*) формирует результат переноса ***c′***, который поступает на вход ***c*** обычного сумматора в разряде 1 (рис. 23, *в*).

Таблица 7

Таблица состояний полусумматора

a_0	b_0	s	c′
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Арифметико-логическое устройство

Арифметико-логическое устройство (АЛУ), как правило, входит в состав любого процессора и выполняет автоматизированную обработку данных. Рассмотрим одноразрядное АЛУ, способное выполнять 4 команды, из которых 3 логические (И, ИЛИ, ИЛИ-НЕ) и одна арифметическая — сложение (рис. 24).

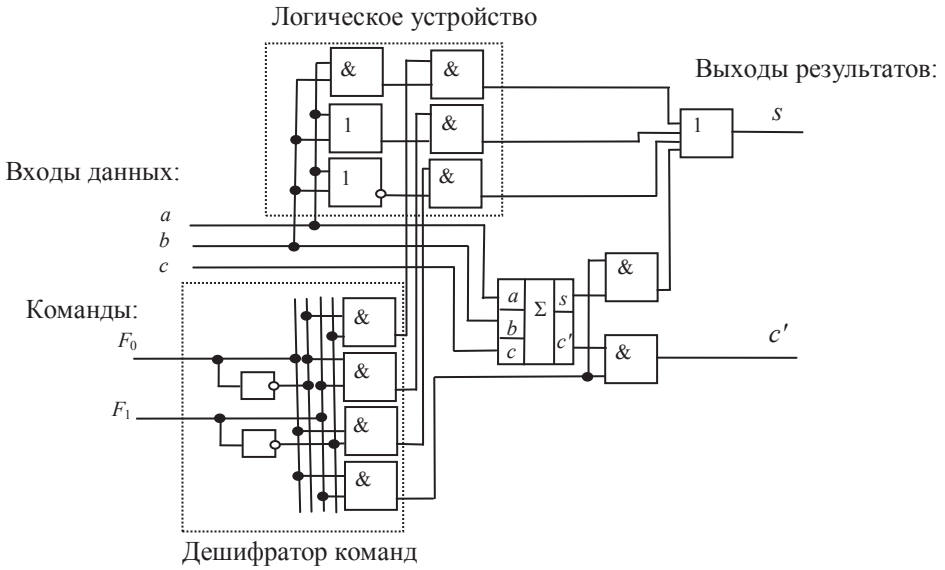


Рис. 24. Одноразрядное арифметико-логическое устройство

Основными элементами АЛУ являются дешифратор команд, логическое устройство и сумматор. Входные сигналы разделяются на входы данных (линии a, b, c) и на входы команд (линии F_0, F_1). В отли-

чие от линий *a* и *b*, которые используются всеми обрабатывающими блоками АЛУ, линия с представляет собой вход переноса и используется только сумматором. Результатом работы АЛУ являются сигналы на линии *s*, а также сигналы на линии с′ в случае выполнения команды сложения. В табл. 8 представлены состояния данного устройства при выполнении логических команд.

Таблица 8

Выполнение логических команд

Команда	F_1	F_0	a	b	s
И	0	0	0	0	0
			0	1	0
			1	0	0
			1	1	1
ИЛИ-НЕ	0	1	0	0	1
			0	1	0
			1	0	0
			1	1	0
ИЛИ	1	0	0	0	0
			0	1	1
			1	0	1
			1	1	1

Рассмотрим работу АЛУ при выполнении команды И. Код команды (00_2) поступает на дешифратор по линиям F_0 и F_1. Распознавание команды сводится к установке единицы на верхней выходной линии дешифратора и нулей на всех остальных линиях (рис. 25).

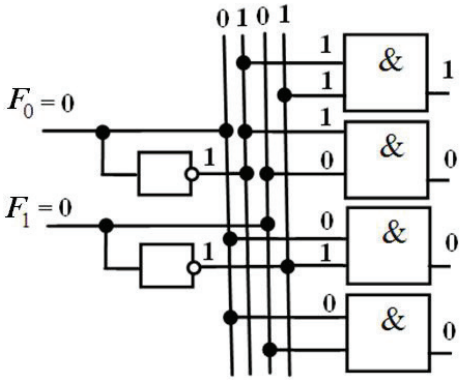


Рис. 25. Дешифрация команды И

Выходные сигналы дешифратора поступают на логическое устройство (рис. 26). Структурно-логическое устройство состоит из двух частей: из элементов, которые выполняют операции над сигналами *a* и *b* и выходных элементов И, которые разрешают или запрещают прохождение сигнала далее по схеме, используя сигнал, полученный от дешифратора команд. В рассматриваемом случае единичный сигнал поступает от дешифратора команд на самый верхний из выходных элементов И. В итоге на элемент ИЛИ, формирующий сигнал на линии *s* (см. рис. 24), поступает результат логической операции И над входными сигналами *a* и *b* (табл. 8).

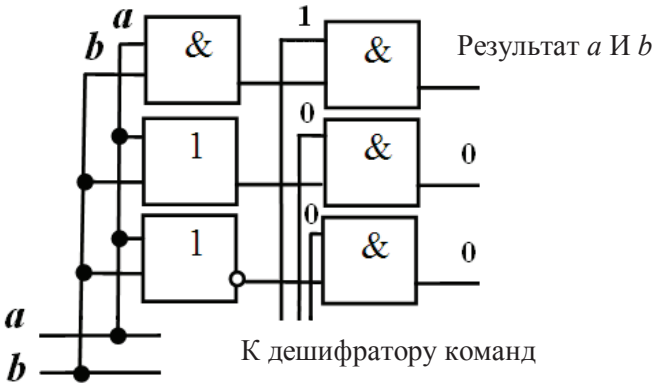


Рис. 26. Работа логического устройства

Аналогичным образом рассматриваемое арифметико-логическое устройство выполняет остальные команды. В качестве самостоятельного упражнения заполните табличную форму для команды сложения:

Команда	**F_1**	**F_0**	**a**	**b**	**c**	**s**	**c′**
Сложение			0	0	0		
			0	0	1		
			0	1	0		
			0	1	1		
			1	0	0		
			1	0	1		
			1	1	0		
			1	1	1		
			1	1	1		

Триггер

Триггер, как уже упоминалось, является основным устройством, которое физически реализует двоичный разряд (бит). Объединением соответствующего числа триггеров получают регистры, в которых размещаются байты, слова и другие элементы данных. Также триггеры составляют основу статической оперативной памяти.

Основу триггера составляет схема, состоящая из двух последовательно включенных инверторов, охваченных обратной связью (рис. 27, *а*). Последнее обстоятельство как раз и обеспечивает рассматриваемой схеме способность принимать одно из двух состояний и находиться в каждом из них при условии сохранения питания неограниченное время. Например, при занулении входа первого инвертора на его выходе сформируется единичный сигнал (см. рис. 14, *б*), который после прохождения второго инвертора примет нулевое значение и попадет на вход схемы. После этого исходный, инициирующий сигнал может быть снят без изменения состояния схемы. При инициирующей установке первого инвертора в единицу, схема аналогичным образом принимает альтернативное состояние.

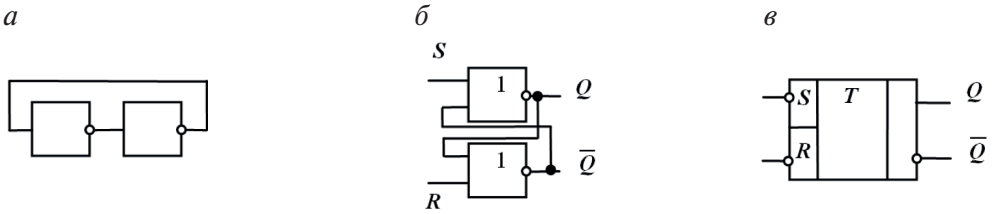


Рис. 27. Асинхронный *RS*-триггер:
а — последовательностная схема; *б* — схема триггера; *в* — условное изображение

Если инверторы разместить друг над другом и для удобства управления заменить каждый из них, например на элемент ИЛИ-НЕ, получим асинхронный *RS*-триггер (рис. 27, *б*). Асинхронным он называется из-за отсутствия отдельного синхронизирующего входа, определяющего периоды работы схемы, в течение которых она способна изменять свое состояние. По этой причине работа рассматриваемого асинхронного триггера определяется только сочетанием сигналов на входах *S* и *R*. Для триггеров принято использовать устоявшиеся названия входов и выходов. В данном случае вход *R* (reset) — это вход сброса (установка триггера в ноль), а вход *S* (setup) — это вход установки триггера

в единицу. Сигналы на выходах триггера всегда должны быть в парафазном (противоположном) коде: если на одном ноль, то на другом единица, и наоборот. Основным выходом считается верхний — Q. По нему определяется состояние всего триггера. Второй выход обозначается как инверсия основного: $\overline{Q}$.

Работу триггера описывает таблица, которую принято называть операционной (табл. 9). В общем случае новое состояние триггера определяется как сигналами на его установочных входах (R и S), так и состоянием Q^n, в котором триггер находился до этого момента. Новое состояние триггера обозначается как Q^{n+1}.

По схеме (рис. 27, *б*) рассмотрим, как получается состояние триггера в первой строке табл. 9. На входе S в данном случае единица, однозначно определяющая нулевое значение на выходе верхнего элемента ИЛИ-НЕ. Этот нуль формирует выход Q, а также подается на вход нижнего элемента ИЛИ-НЕ. Вторым входом для этого элемента является вход R также со значением ноль, поэтому на инверсном выходе триггера формируется единичное значение. Аналогичные рассуждения справедливы и для второй строки табл. 9.

Таблица 9

Операционная таблица *RS*-триггера на элементах ИЛИ-НЕ

R	S	Q^{n+1}
0	1	0
1	0	1
0	0	Q^n
1	1	Запрещенное состояние

В третьей строке табл. 9 описывается состояние триггера, в котором он осуществляет хранение данных (Q^n-сигнал на выходе не меняется). Рассмотрим это подробнее. По нулевым сигналам на входах S и R (рис. 27, *б*) невозможно определить, какие сигналы формируют элементы ИЛИ-НЕ на своих выходах. Дополнительной информацией в данном случае служат данные о текущем состоянии триггера. Предположим, что триггер находился в единичном состоянии: на прямом выходе была сформирована единица, на инверсном — ноль. Тогда единица с выхода Q подается на вход нижнего элемента ИЛИ-НЕ и определяет на его выходе нулевое значение. В свою очередь этот нуль по цепи обратной связи подается на верхний элемент ИЛИ-НЕ и вместе с нулевым значением входа S формирует на выходе этого элемента еди-

ничное значение. Круг замкнулся: предположив единичное состояние триггера, мы убедились, что сигналы на установочных входах триггера это состояние не изменяют. К похожим выводам можно прийти, предположив то, что триггер изначально находился в нулевом состоянии.

Состояние триггера, описанное в последней строке табл. 9, отмечено как запрещенное (нерабочее). Причина в том, что единичные значения сигналов на входах *R* и *S* переключают оба элемента ИЛИ-НЕ в нулевое состояние, нарушая парафазность выходов триггера.

Несложно заметить (табл. 9), что установочные входы асинхронного *RS*-триггера работают с точность до наоборот: единица на установочном входе сбрасывает триггер в нулевое состояние (строка 1), а единица на входе сброса устанавливает триггер в единичное состояние (строка 2). По этой причине на условном изображении триггера входы *S* и *R* обозначены входной инверсией (рис. 27, *в*).

Конструкция и основы функционирования процессора Intel8080

Выпуск процессора Intel8080 фирмой Intel был начат в далеком 1974 году. С 1976 года фирма начала выпуск несколько улучшенной версии процессора под маркой Intel8085, который использовался вплоть до 2000-х годов. На базе удвоенной версии этого процессора в 1979 году был начат выпуск шестнадцатиразрядного устройства Intel8086, послужившего основой для персональных компьютеров. В итоге многие из реализованных в Intel8080 архитектурных решений получили свое развитие и остаются востребованными в современных информационных системах.

Процессор относится к классу восьмиразрядных, так как размер основных регистров составляет один байт. Это регистры общего назначения *A*, *B*, *C*, *D*, *E*, *H*, *L*, регистр флагов *F* и два служебных rg*A* и rg*B* (см. рис. 10 на с. 23). Регистры общего назначения используются, как правило, для подготовки данных перед их обработкой сумматором. Особая миссия при этом у регистра А — аккумулятора. В нем в обязательном порядке размещается один операнд команды, а также аккумулятор используется для размещения результата операции, выполненной сумматором (рис. 10). При этом признаки этого результата сохраняются в соответствующих разрядах регистра *F* (рис. 28). Здесь *S* (signal) — флаг знака, *Z* (zero) — флаг нуля, *A* (additional carry) — флаг

дополнительного переноса, *P* (parity) — флаг четности, *C* (carry) — флаг переноса.

Регистры *B* и *C*, *D* и *E*, *H* и *L* можно использовать и парами, что дает возможность выполнять некоторые действия сразу с шестнадцатиразрядными данными (см. рис. 10, с. 30).

Номер разряда	7	6	5	4	3	2	1	0
Обозначение флага	*S*	*Z*	–	*A*	–	*P*	–	*C*

Рис. 28. Регистр флагов

Регистры *PC* (program counter — программный счетчик) и *SP* (stack pointer — указатель стека) имеют нестандартный размер в два байта и предназначены для работы с устройствами памяти. Содержимое регистра *PC* — это адрес команды, которую процессор должен выполнить следующей. Содержимое *SP* — адрес вершины стека, специального участка оперативной памяти, работающего по принципу обратной очереди (LIFO — last in first out, последний пришедший выходит первым). Наличие этих регистров обеспечивает процессору возможность работы с устройствами памяти размером до 64 К (2^{16}).

В автоматизированную систему процессор объединяется с помощью 8-разрядной шины данных и 16-разрядной шины адреса. При этом младшие восемь разрядов шины адреса могут быть использованы для адресации портов, выполняющих роль интерфейсов с внешними устройствами. Таким образом, потенциально процессор может работать с 256 устройствами (2^8).

Как уже отмечалось, задача процессора — обработка данных под управлением соответствующей программы. Для этого перед началом работы программа должна быть загружена в устройство памяти, а адрес ее первой команды находиться в регистре PC (см. рис. 10). После этого процессор:

1) осуществляет считывание команды в регистр КОП (кода операции);
2) с помощью DC КОП (дешифратора кода операции) осуществляет ее декодирование;
3) если текущая команда — команда останова, прекращает выполнение программы, если любая другая команда — выполняет п. 4;

4) в соответствии с длиной команды выполняет коррекцию содержимого *PC*, чтобы настроить его значение на адрес следующей команды;
5) при необходимости обращается к устройству памяти для получения дополнительных данных для текущей команды;
6) выполняет команду с помощью сумматора, формируя результат (в регистре А) и признаки результата (в регистре *F*);
7) продолжает работу с п. 1.

В составе процессора Intel8080 почти 5 тысяч транзисторов и работает он на частоте 2,5 Мгц.

Назначение и краткая история развития устройств памяти

Устройства памяти предназначены для кратко- или долговременного хранения кода (или команд) и данных в виде последовательности двоичных значений и делятся на *внешнюю* и *внутреннюю* память.

Внешняя память реализуется в виде отдельных устройств (магнитные, оптические диски, флэш-накопители и др.). Доступ к данным осуществляется поблочно и через промежуточный буфер. Данные устройства характеризуются большими объемами хранимых данных (сотни и тысячи Гбайт), значительным временем доступа, небольшой удельной стоимостью и, как правило, возможностью достаточно длительного хранения данных.

Внутренняя память компьютеров обычно не превышает размера в несколько десятков Гбайт, имеет малое время доступа, но при этом достаточно большую удельную стоимость хранения данных. В то же время доступ к данным осуществляется побайтно и непосредственно.

В первых компьютерах, выполненных на электромагнитных реле, память использовалась только для хранения данных, подготовленных для обработки и записи результатов вычислений. Объема такой памяти хватало для размещения достаточно ограниченного количества чисел. В большей степени такая память выполняла роль регистров современных компьютеров. Первое, как таковое, устройство памяти было использовано в ламповом компьютере Эниак и имело объем, достаточный для хранения лишь 20 десятиразрядных десятичных чисел.

Примерно в то же время математиком Джоном фон Нейманом было доказано, что увеличение быстродействия компьютера произойдет только тогда, когда наряду с числами и сама программа будет храниться

в устройстве памяти. Практическая реализация этого принципа потребовала создания устройств памяти гораздо большего объема. Использовать напрямую для этого электронные лампы оказалось невозможным по нескольким причинам: дорого, громоздко, большие затраты электроэнергии и проблема отвода тепла.

В 1948 году учеными Манчестерского университета было предложено использовать в качестве устройств памяти электронно-лучевые трубки. К фосфорному покрытию кинескопа подключались электроды, выведенные наружу. Изнутри по покрытию с периодичностью 30 раз в секунду перемещался электронный луч, и засвеченные точки получали электрический заряд, который и регистрировался впаянными в кинескоп электродами. Для обеспечения работоспособности компьютера в минимальном объеме требовалось не менее шести таких трубок. За последующие годы устройства памяти выполнялись на ртутных линиях задержки, на ферритовых кольцах и сердечниках. Наконец, в 1968 году фирма Intel объявила о начале серийного выпуска первой интегральной полупроводниковой микросхемы памяти объемом информации в 1 Кбит.

Виды внутренней памяти

По принципу действия современные устройства внутренней памяти делятся на два типа: *ROM* (*read only memory* — память только для чтения) или постоянное запоминающее устройство (ПЗУ) и *RAM* (*random access memory* — память произвольного доступа), или оперативное запоминающее устройство (**ОЗУ**).

Основным преимуществом *ROM* является энергонезависимый режим хранения информации. *RAM* является энергозависимым устройством: при отключении питания информация теряется без возможности восстановления. При этом *RAM* может работать в режимах и чтения, и записи данных.

Постоянное запоминающее устройство

Существует несколько видов *ROM*-памяти. Самое высокое быстродействие имеют масочные *ROM*. Структурно устройство памяти состоит из матрицы запоминающих элементов (МЗЭ) и дешифратора адреса (рис. 29).

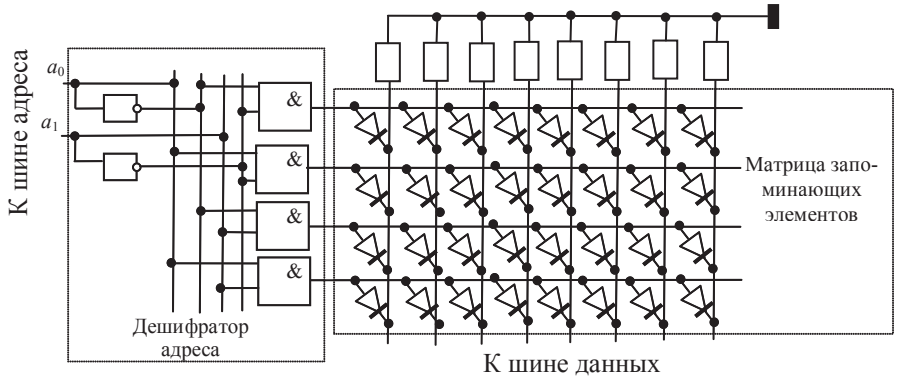


Рис. 29. Масочная *ROM* 4×8 на первом этапе изготовления

МЗЭ содержит строки по количеству ячеек памяти и столбцы — по размерности хранимых данных. Ячейка матрицы определяется пересечением выходной линии дешифратора и вертикальной линии, подключаемой к разряду шины данных (рис. 29). В качестве примера предположим, что создается *ROM* объемом 4 ячейки по 8 бит или, как говорят, формата 4 на 8. На первом этапе изготовления каждая строка матрицы запоминающих элементов связывается с каждым столбцом отдельным диодом. В итоге получается, что устройство памяти заполнено единичными значениями.

На втором этапе с учетом информации, которую необходимо разместить в устройстве памяти, подготавливают маску — непрозрачный экран, имеющий отверстия над теми диодами, которые из матрицы требуется удалить.

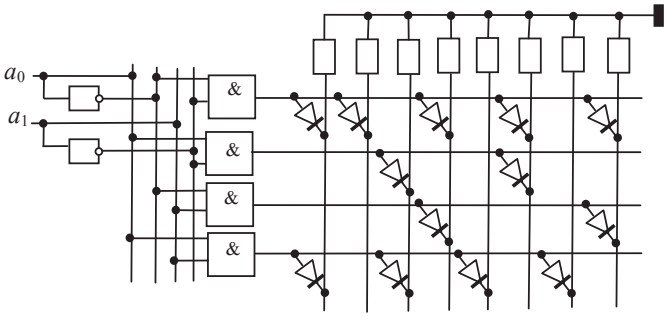
Третий этап изготовления — программирование матрицы. Для этого МЗЭ накрывают маской и обрабатывают пучком высокой энергии (электронный или лазерный луч). Диод, попавший под отверстие маски, выгорает и соответствующая связь между горизонтальной и вертикальной линиями МЗЭ исчезает. На рис. 30 показана диодная матрица после выжигания диодов.

Коротко о работе схемы. В соответствии с сигналами на шине адреса (линии a_1 и a_0) дешифратор адреса подает потенциал 5 В на одну из своих выходных линий (рис. 30, *а*). Диоды выбранной линии оказываются включенными в прямом направлении и передают на линии данных потенциал единицы. Те линии данных, которые не имеют электрической связи с выбранной линией адреса (диод отсутствует),

получают 0 В (потенциал земли) за счет подтягивающих сопротивлений в верхней части схемы.

Содержимое памяти удобно представлять в виде карты памяти. В левой колонке указывается адрес ячейки, в правой — ее содержимое, как правило, в двоичной системе (рис. 30, б).

а



б

Адрес	Содержимое
0	11010101
1	00100100
2	00010001
3	10101010

Рис. 30. Масочная *ROM* 4×8:

а — микросхема памяти после программирования; *б* — карта памяти

В качестве самостоятельного упражнения сформируйте карту памяти с содержимым в двоичной и шестнадцатеричной системах для устройства, схема которого приведена на рис. 31.

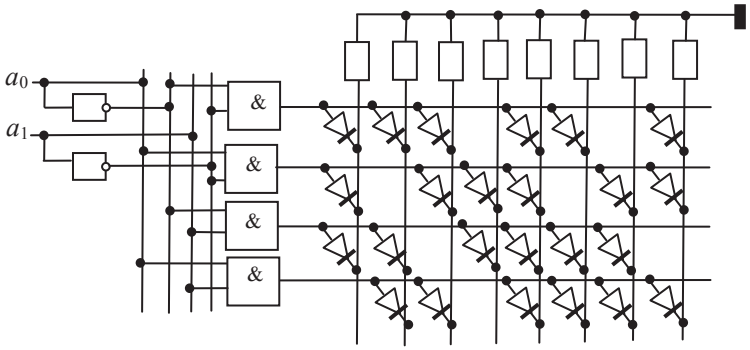


Рис. 31. *ROM* 4×8

Масочная *ROM*-память остается одной из самых быстродействующих. Однако такие ее недостатки, как однократность программирования и необходимость специального прецизионного (высокоточного) оборудования ограничивают ее применение случаем массового промышленного производства.

Возможность программирования практически в домашних условиях предоставляют устройства, выполненные по *PROM* технологии (Programming *ROM* — программируемая *ROM*) прежде всего, за счет иной схемы запоминающего элемента. В данном случае в цепи, соединяющей линии адреса с линиями данных последовательно с диодом, включается высокоомная перемычка, получаемая напылением нихрома (рис. 32, *а*).

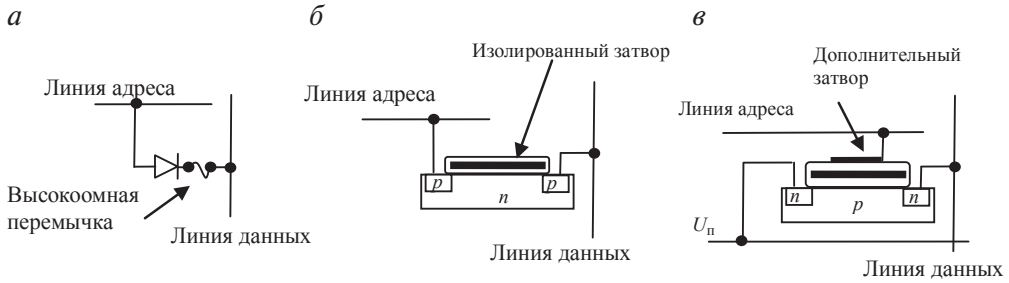


Рис. 32. Запоминающие элементы *PROM*-памяти:
а — PROM; *б — EPROM*; *в — EEPROM*

Схема запоминающего устройства *PROM* представлена на рис. 33. В процессе изготовления формируется заготовка, все запоминающие элементы которой содержат единичные значения (рис. 33, *а*). Для записи нулей в соответствующие ячейки МЗЭ используется специальное устройство — программатор. С его помощью выбирается запоминающий элемент, затем между линией адреса и линией данных подается повышенное напряжение. Ток, протекающий через цепь диод, — перемычка, нагревает и распыляет нихром.

В качестве самостоятельного упражнения сформируйте карту памяти с содержимым в двоичной и шестнадцатеричной системах для микросхемы *PROM* по рис. 33, *б*.

Однократность программирования — основной недостаток *ROM* и *PROM* технологий, преодоленный в *EPROM* (*erasable PROM* — стираемой программируемой *ROM*-памяти). Запоминающим элементом такой памяти является полевой транзистор с изолированным затвором, который размещен внутри изолятора и не имеет собственного вывода (рис. 32, *б*). В заготовке *EPROM*-памяти каждая линия адреса МЗЭ соединяется с каждой линией данных подобным транзистором, и получается, что во все запоминающие элементы записаны нули, так как на изолированном затворе отсутствует какой-либо электрический заряд

и транзистор находится в закрытом состоянии. Для записи единичного значения на выбранный запоминающий элемент с помощью программатора подается импульс повышенного напряжения, и изоляционные промежутки между *p*-областями и изолированным затвором транзистора пробиваются. После спада импульса электрическая прочность промежутка восстанавливается. При этом ток пробоя успевает зарядить изолированный затвор подобно пластине конденсатора до некоторого отрицательного потенциала, тем самым обеспечивая открытое состояние транзистора. В данном состоянии за счет сохранности заряда изолированного затвора полевой транзистор может находиться десятки лет, связывая своей проводимостью соответствующие линии адреса и линии данных и формируя при соответствующем обращении к запоминающему элементу единичный уровень сигнала на его выходе.

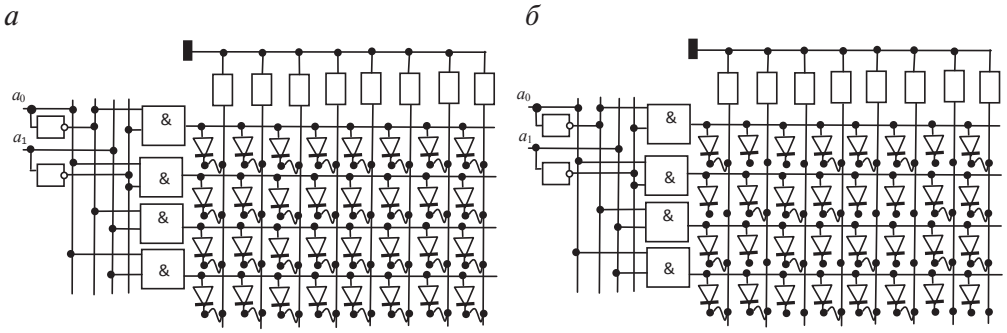


Рис. 33. Память *PROM*:
а — заготовка для микросхемы *PROM* 4×8;
б — микросхема *PROM* 4×8 после программатора

Перед повторным программированием микросхемы *EPROM* необходимо стереть все ранее записанные единицы. Для этого микросхему вынимают из платы, где она используется, и помещают под источник ультрафиолетового излучения (сверху микросхемы для этого предусмотрено специальное прозрачное окошко). Во время облучения электроны, обеспечивающие отрицательный заряд затвора, получают квант световой энергии и, преодолевая потенциальный барьер изоляционного промежутка, разряжаются на подложку. Проводящий канал рассасывается, транзистор закрывается, на линии данных при соответствующем обращении формируется нулевой потенциал.

Неудобства, связанные с необходимостью изъятия микросхемы *EPROM* из платы для ее перепрограммирования, преодолены

в *EEPROM* (*electrical erasable* — электрически стираемая программируемая *ROM*). В качестве запоминающего элемента в данном случае используется двухзатворный *n*-канальный полевой транзистор (рис. 32, *в*). Один из его затворов так же, как и в предыдущем случае, изолированный, а второй, дополнительный, — вполне обычный. При этом в корне меняется роль изолированного затвора: в *EPROM* его заряд открывает транзистор, а в *EEPROM* отрицательный потенциал изолированного затвора противодействует полю положительного потенциала дополнительного затвора в попытке индуцировать канал *n*-типа и открыть транзистор.

Таким образом, заготовка *EEPROM* заполнена единицами, а для записи нулей необходимо изолированному затвору сообщить экранирующий отрицательный заряд. Для этого с помощью программатора выбирают нужный транзистор, на его линии данных обеспечивают нулевой потенциал, а на линию $U_п$ и дешифратор подают импульс повышенного напряжения.

Для восстановления единицы (разряда плавающего затвора) повышенное положительное напряжение подается только на дешифратор.

Широко используемые в настоящее время флэш-накопители являются разновидностью *EEPROM*. За счет отказа от возможности побайтного стирания ранее записанной информации (стирание возможно только блоками) у флэшек существенно упрощается схема, увеличивается плотность данных и, самое главное, — объем хранимой информации.

Оперативное запоминающее устройство

Оперативная память может быть выполнена на основе *статических* или *динамических* элементов.

Статическая память (*SRAM — static RAM*) использует в качестве запоминающего элемента триггер, при этом является более сложной, энергоемкой и дорогой. В среднем на один разряд здесь приходится 8—10 транзисторов. Основное преимущество *SRAM* — отсутствие служебного режима, во время которого запоминающий элемент недоступен. Термин статическая, характеризующий этот вид памяти, означает, что информация в подобных устройствах сохраняется при снижении тактовой частоты, на которой работает информационная система,

вплоть до нулевых значений. Единственным условием штатной работы *SRAM* является наличие напряжения питания (рис. 34).

Статическая память представляет собой двумерную структуру, каждый запоминающий элемент которой выбирается по номеру строки и номеру столбца. Выбор ЗЭ реализует элемент И, на который подаются сигналы соответствующих линий (рис. 34). Сам триггер собран на транзисторах *VT*3 и *VT*4, транзисторы *VT*1 и *VT*2 — это электронные ключи, обеспечивающие проводимость в режиме, когда данный запоминающий элемент выбран.

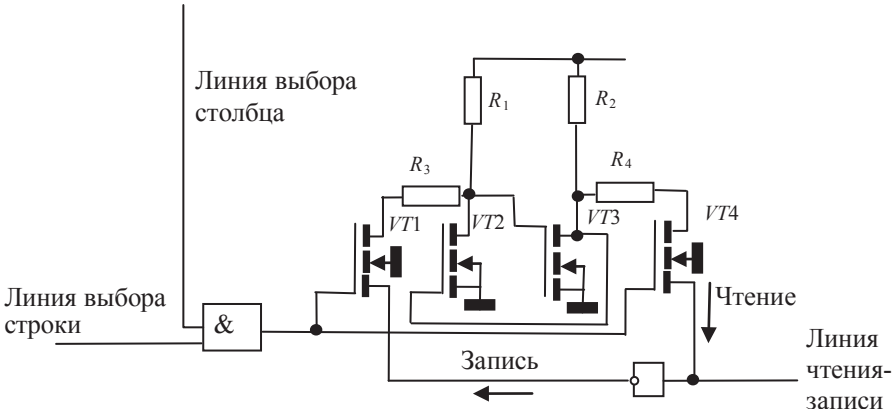


Рис. 34. Запоминающий элемент *SRAM*

Динамическая память (*DRAM — dynamic RAM*) использует в качестве запоминающего элемента бесстоковый полевой транзистор, обладающий электрической емкостью между областями стока и подложки (рис. 35, *а*). Эта емкость ($C_{зэ}$ на схеме) выполняет роль запоминающего элемента, а остальная часть безстокового транзистора (затвор и канал *n*-типа) выполняет роль электронного ключа (рис. 35, *б*). $C_{зэ}$ имеет очень небольшой номинал, сопоставимый с емкостью линии записи-считывания, объединяющей соответствующие разряды МЗЭ. По этой причине электрический заряд $C_{зэ}$ после выбора ЗЭ и открытия транзистора не может обеспечить уровень напряжения на линии записи-считывания, соответствующий единичному уровню сигнала. Решается данная проблема с помощью предварительного заряда линии записи-считывания (перед каждым обращением к ЗЭ) до половины напряжения питания. В момент подключения емкости $C_{зэ}$ усилитель — регенератор регистрирует знак изменения напряжения на линии записи-считыва-

ния. Если напряжение увеличилось, в запоминающем элементе была записана единица, в противном случае — ноль. Усилитель-регенератор формирует на линии данных сигналы установленных уровней напряжения, а также при необходимости восстанавливает заряд $C_{зэ}$.

Небольшая емкость элемента и токи утечки требуют постоянной регенерации (перезаряда) $C_{зэ}$ с периодичностью от 8 до 64 мс. Отличается *DRAM*-память сравнительно небольшой стоимостью, но несколько замедляет работу компьютера, т. к. во время регенерации любые операции чтения-записи запрещены. На основе *DRAM*-элементов выполняются основная память и модули расширения.

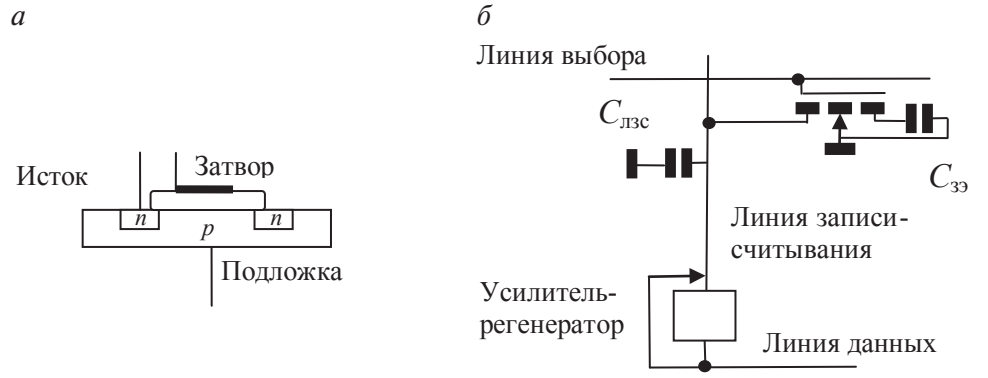


Рис. 35. Запоминающий элемент *DRAM*:
а — безстоковый транзистор; *б* — схема запоминающего элемента

Автоматизированная система на базе Intel8080

Как уже упоминалось, в составе системы, позволяющей под программным управлением выполнять обработку данных, в обязательном порядке присутствуют процессор, устройства *RAM* и *ROM*-памяти, внешние устройства, схемы, обеспечивающие их взаимодействие (порты, схемы прерываний и прямого доступа), а также шины адреса и шины данных, непосредственно осуществляющие информационный обмен (рис. 36).

После включения питания, а также после сигнала сброса программный счетчик обнуляется и поэтому процессор первой выполняет команду *Rst*0 (нулевого рестарта), которая находится в самом начале адресного пространства, по нулевому адресу. Как правило, эта команда осуществляет переход на специальную программу в составе опера-

ционной системы (ОС), которая обеспечивает, прежде всего, взаимодействие с пользователем (программа — монитор). Кроме монитора, в составе ОС имеются программы ввода с клавиатуры, вывода на монитор и др.

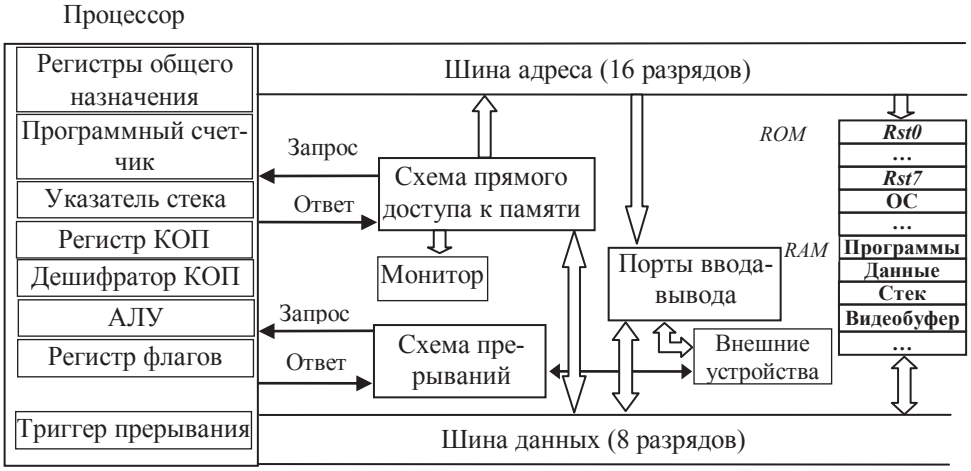


Рис. 36. Структура автоматизированной системы на базе Intel8080

Взаимодействие процессора с внешними устройствами может быть организовано либо с помощью опроса, либо с использованием прерываний. В первом случае инициатива обмена данными принадлежит процессору. Он обращается к внешнему устройству с вопросом о готовности к обмену и в случае положительного ответа организует обмен через шины адреса и данных, а также порты ввода — вывода. Издержками этого способа обмена являются нерационально большие затраты основного ресурса системы — процессорного времени.

При использовании прерываний инициатива обмена делегирована внешним устройствам. По мере готовности внешнее устройство через схему прерываний передает запрос процессору. Для процессора это означает, что выполнение текущей программы необходимо прервать и переключиться на специальную программу, которая осуществляет обмен данными с запросившим прерывание устройством. При этом процессор должен позаботиться о том, чтобы после завершения обмена он смог вернуться к прерванной программе, и это никак не повлияло бы на результаты ее работы. Возврат к прерванной программе обеспечивает сохранение в стеке содержимого программного счетчика.

Как только процессор закончит выполнение команды, во время которой поступил запрос прерывания, и сохранит адрес возврата в стеке, он формирует ответ о готовности переключиться на работу с внешним устройством. После этого внешнее устройство по шине данных сообщает процессору, на какую из программ рестарта (*Rst*1 — *Rst*7) он должен переключиться, чтобы выполнить обмен.

Таким образом, при использовании прерываний процессор основное время выполняет полезную работу, отвлекаясь на обмены только по мере необходимости. Процессору можно запретить реагировать на запросы прерывания, если специальной командой обнулить триггер прерывания (рис. 36).

Особое место в рассматриваемой системе занимает монитор, работа которого организована через схему прямого доступа к памяти (ПДП). С периодичностью, обеспечивающей неразличимую человеческим зрением пульсацию изображения на мониторе, схема ПДП обращается с запросом к процессору. Получив запрос, процессор практически сразу должен отключиться от шин адреса и данных, сформировать ответ и оставаться в режиме ожидания. За время монопольного владения шинами устройство ПДП осуществляет считывание из *RAM*-памяти содержимое видеобуфера и воспроизводит по нему изображение на мониторе. После этого запрос снимается и процессор продолжает работать до следующего сеанса обновления монитора. Следует заметить, что само изображение может формировать любая программа, так как видеобуфер является частью *RAM*-памяти (рис. 36).

Программное обеспечение

Совокупность программ называют программным обеспечением [1]:

- системное программное обеспечение — специальные программы, обеспечивающие работу аппаратной части компьютера;
- универсальное программное обеспечение, прежде всего, приложения пакета *Microsoft Office*, программы для работы с общепризнанными форматами документов (ридеры), браузеры и другие средства для работы в сети интернет;
- прикладное программное обеспечение — программы для решения технологических задач, например расчета установившегося режима, расчета токов короткого замыкания и др.

Основной частью системного программного обеспечения являются операционные системы (ОС) — это комплекс программ, управляющих ресурсами компьютера. Под ресурсами понимают процессорное время, объем оперативной памяти, периферийные устройства. Управление предполагает организацию доступа и эффективное распределение этих ресурсов между, как правило, параллельно работающими программами (процессами).

Также операционная система обеспечивает удобную работу с периферийными устройствами компьютера. Для именования накопителей выбраны буквы латинского алфавита с обязательным символом «двоеточие» в конце. При этом *A*: и *B*: — это всегда накопители на гибких магнитных дисках, которые в настоящее время, практически, не используются. Накопители на жестких магнитных и оптических дисках, а также флэш-накопители могут иметь имена *C*:, *D*:, *E*: и далее, вплоть до конца алфавита.

Логическая структура накопителей представляет собой иерархическую, древовидную систему каталогов и файлов (рис. 37). Файл — это набор, как правило, однотипных данных, размещенный на дисковом устройстве и имеющий имя.

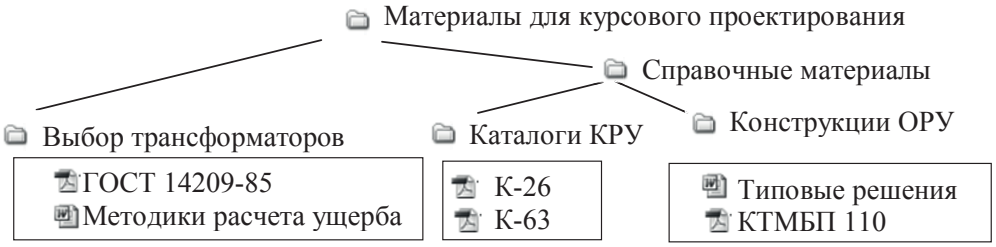


Рис. 37. Дерево файловой системы

Ранее, в операционной системе *MS-DOS*, имя файла формировалось только из латинских букв по правилу «8.3»: — до 8 символов — основное имя, до 3 символов — расширение. ОС *Windows* позволяет использовать имена файлов, каталогов и ярлыков, содержащие до 255 символов. Преемственность осталась в именовании расширения: как правило, более трех-четырех символов оно не содержит. В локализованных версиях операционной системы допускается использовать для именования элементов файловой системы символы национальных алфавитов.

Само расширение представляет собой информацию о типе данных, которые содержатся в файле. Наиболее распространенные расширения:

exe, com — исполняемые файлы, которые содержат подготовленные к выполнению программы;

bat — командный файл, который содержит команды ОС;

txt — текстовый файл, который содержит коды символов;

sys — системный файл, который содержит компоненты ОС;

bak — «старая копия», которая содержит данные в состоянии до последней коррекции;

doc — документ текстового редактора *Microsoft Word*;

xls — документ табличного процессора *Microsoft Excel*;

mdb — база данных *Microsoft Access*.

Каталог — это список из имен файлов и имен каталогов нижерасположенного уровня (подкаталогов). Древовидной файловая система называется из-за отсутствия поперечных связей между иерархическими уровнями. Путь к файлу вместе с именем файла (*pathname*) всегда уникален, начинается с корня диска и является единственной координатой файла. Обратная косая черта в пути к файлу используется для разделения уровней, например: Материалы для курсового проектирования\Выбор трансформаторов\Методики расчета ущерба.doc (рис. 37).

Каждая из последующих глав представляет собой краткое описание и набор этапов, прохождение которых обеспечивает освоение той либо иной технологии обработки данных.

3. Операционная система Windows

Процесс загрузки операционной системы Windows начинается автоматически после включения компьютера. Работа в операционной системе сводится к управлению объектами. Основные объекты Windows представлены на рис. 40: окно, значок, главное меню, панель задач, особая кнопка — Пуск и др. Перечислим основные приемы работы с объектами:

- «Нажать кнопку» — навести указатель (курсор) мыши на изображение кнопки и один раз нажать на левую клавишу;
- «Щелкнуть на объекте» — навести указатель мыши на объект и один раз нажать основную (левую) клавишу мыши. Таким способом осуществляется выбор нужного объекта (при этом его изображение будет выделено системой). Будет специально оговорено, если «щелкнуть» необходимо правой кнопкой мыши;
- «Вызвать контекстное меню» — оставаясь указателем мыши в границах объекта, нажать правую клавишу мыши. Контекстное меню содержит основные команды редактирования (например, копирование, вставка) и настройки объекта;
- «Активизировать объект» — выполнить двойное нажатие основной (левой) клавиши мыши, находясь указателем на выделенном объекте;
- «Перемещение (перетаскивание) объектов» — выделить объект, и, не отпуская левой клавиши мыши, начать перемещение указателя в нужное место. То же самое можно достигнуть и при нажатии правой клавиши мыши, а режим перетаскивания определить с помощью появляющегося при отпускании клавиши меню.

Первое знакомство

К началу вашей работы операционная система Windows должна быть загружена на компьютере. Ознакомьтесь с внешним видом рабочего стола. Найдите на экране следующие объекты: курсор мыши, Панель задач, папки, значки, кнопки.

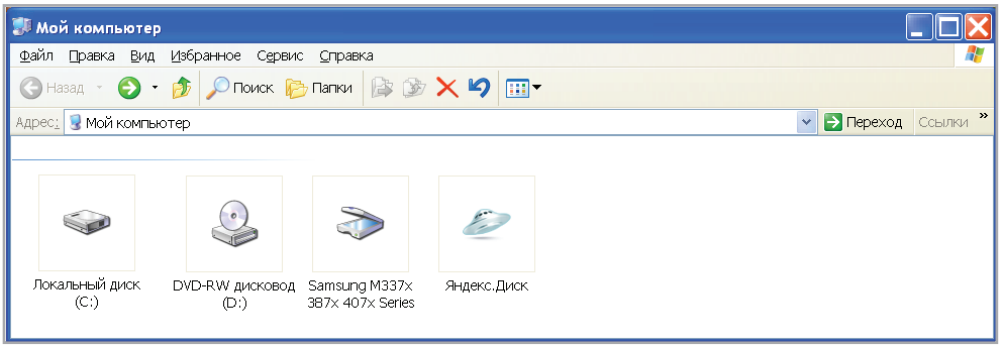


Рис. 38. Окно системной папки *Мой компьютер*

Панель задач, расположенная обычно вдоль нижней границы рабочего стола, является специальным средством Windows для быстрого доступа к программам и документам. Сразу после запуска системы *Панель задач* содержит кнопку *Пуск*, индикатор клавиатуры и, возможно, часы (в правой части панели). Кнопка *Пуск* имеет единственное назначение — открывать *Главное меню*. Запуск каждого приложения добавляет соответствующую кнопку на панель. Кнопки *Панели Задач* удобно использовать для быстрого переключения между приложениями или документами.

Этап 1

- Изучите *Рабочий стол*, установите тип и назначение объектов, размещенных на *меню*
- Изучите состав *Главного меню* операционной системы Windows.

Работа с папками

Папка Windows — это графическое представление каталога, хотя некоторые папки содержат принтеры и диски, которые ни файлами, ни подкаталогами не являются. Программа *Мой компьютер* демон-

стрирует содержимое дисков и папок локального компьютера, а программа *Сетевое окружение* — объекты локальной сети. Рабочим диском в компьютерном классе следует считать диск *D*:.

Этап 2

- Просмотрите содержимое *Мой компьютер* и *Сетевое окружение*.
- Создайте новую папку с именем вашей учебной группы на рабочем диске. Если такая папка уже есть, данный пункт можно пропустить.
- Создайте в папке группы личную папку, назвав ее вашей фамилией и инициалами.

Для этого щелкните дважды левой кнопкой мыши на значке программы *Мой компьютер*, из появившегося набора устройств выберите рабочий диск и повторите то же самое еще раз. Когда в рабочем окне появится содержимое рабочего диска, оставаясь на его поверхности курсором мыши, нажмите правую кнопку (вызов контекстного меню), выберите команду *Создать* и в открывшемся подменю — тип создаваемого объекта, т. е. папку. Когда в рабочем поле окна появится новая папка и ее название будет выделено (при этом вы увидите «курсор ввода» — вертикальную пульсирующую черту), наберите нужное название и нажмите клавишу *Enter*.

- Откройте вашу личную папку. Дважды щелкните на значке папки, чтобы открыть ее. Познакомьтесь с элементами окна (рис. 39): значок папки (он же *по совместительству* — кнопка меню управления окном — верхний левый угол окна), строка названия (самая верхняя), кнопки управления окном (правый верхний угол), меню (вторая строка окна).
- Познакомьтесь с командами меню окна папки, измените видимость адресной строки. Сделать это можно в меню *Вид*, командой *Панель инструментов*.
- Создайте внутри папку *Программы*. Зайдите в папку. Открылось при этом новое окно? Если нет, то настройте свойства системной папки таким образом, чтобы содержимое каждой вновь открываемой папки демонстрировалось в новом окне. В меню *Сервис* найдите *Свойства папки* и на закладке *Общие* в разделе *Обзор папок* нажмите левую клавишу мыши на радио-кнопке — напротив поля *Открывать каждую папку в отдельном окне*. Проверьте, как изменилась работа программы.

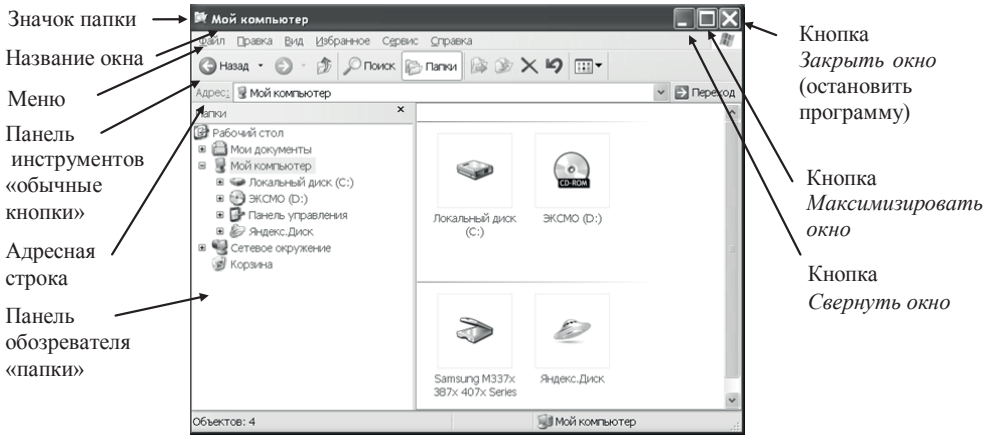


Рис. 39. Основные элементы окна

- Откройте папку *Мой компьютер* и измените форму представления содержимого папок. Сделать это можно с помощью команд в меню *Вид — Эскизы страниц*, *Плитка*, *Значки, Список*, *Таблица*. Установите, какой из режимов является наиболее информативным? Какие характеристики файлов и папок демонстрируются в окне?
- Скопируйте программу *Калькулятор* из каталога *P*:*LAR* в папку *Программы*. Откройте папку *Мой компьютер*, затем папку диска *P*: и саму папку *LAR*. Откройте еще одно окно для папки *Программы*. Найдите в папке *LAR* значок *Calc* (программа калькулятор), перетащите его правой кнопкой на окно папки *Программы* и при отпускании кнопки мыши выберите в появившемся контекстном меню пункт Копировать.

Создание ярлыков

Ярлыки в Windows — это графическое обозначение ссылок на объекты (файлы программ или документов), реально расположенные в другом месте. Основным содержимым ярлыка является полный путь до файла или папки.

Этап 3

- В папке *Программы* создайте ярлык для программы *Calc.exe*, расположенной в *P*:*LAR* (несмотря на то, что в данной папке уже

находится сам объект — программа *Калькулятор*). Сравните характеристики объекта и ярлыка (команда *Свойства контекстного меню*). Измените название ярлыка (с помощью команды *Переименовать* контекстного меню).

Работа с окнами

Окно — это основной элемент (или объект) системы Windows. По определению — это ограниченная рамкой поверхность экрана. Различают окно приложения (открывается при запуске программы), окно документа (таких окон может быть открыто несколько) и диалоговое окно — вспомогательное, предназначенное для установки параметров.

Окно приложения может существовать в трех формах:

- в минимизированном виде кнопкой на панели задач;
- в нормальном размере;
- в полноэкранном варианте.

Для окна документа, в принципе, также доступны перечисленные формы представления, но в минимизированном виде оно имеет форму значка, размещенного, как правило, в границах окна приложения. Форма представления диалогового окна одна — в нормальном размере (рис. 40).

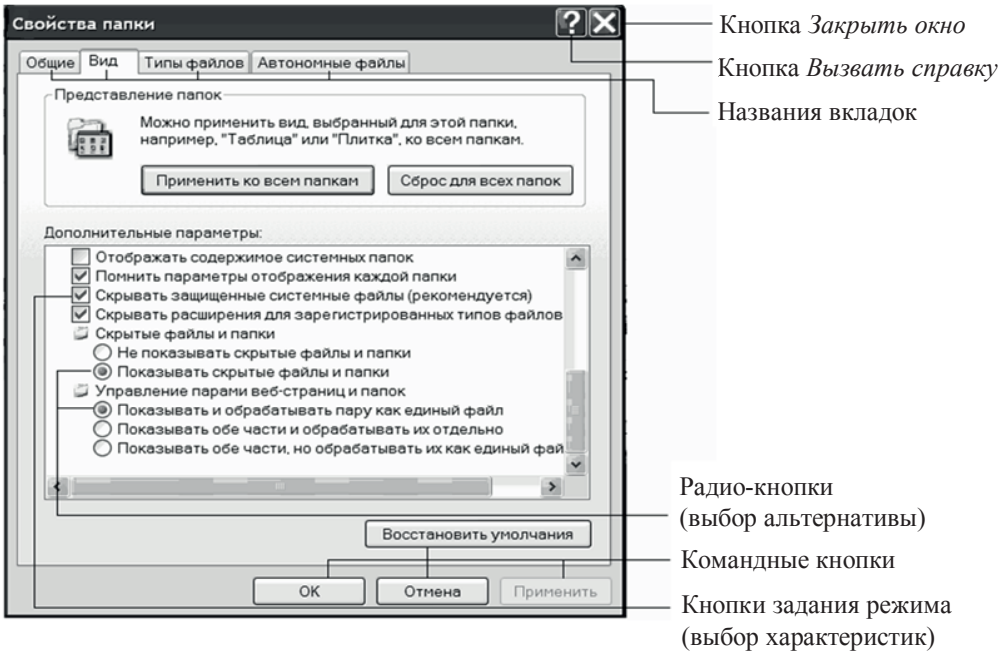


Рис. 40. Диалоговое окно *Свойства папки*

Этап 4

- Проверьте соответствие характеристик программы *Мой компьютер* по рис. 40. Откройте папку *Мой компьютер* и в меню *Сервис* выберите *Свойства папки* и вкладку *Вид*.
- Переместите окно папки на новое место. Для этого откройте папку *Мой компьютер*. Минимизируйте размер окна. Поместите острие указателя внутрь области заголовка окна *Мой компьютер* (первая строка окна). Нажмите и не отпускайте левую кнопку мыши, а затем переместите мышь. При этом будет перемещаться рамочка окна. Отпустите кнопку мыши для размещения окна на новом месте.
- Измените размеры окна**.** Подведите курсор к одной из границ окна (нижней правой), он должен принять вид двойной стрелки, и перемещайте мышь, не отпуская левую клавишу до достижения окном нужного размера.
- Сверните окно *Мой компьютер* в значок. Щелкните на кнопке сворачивания окна *Мой компьютер* (расположена в правом верхнем углу). При этом окно свернется и появится в виде значка на панели задач.
- Восстановите окно из значка. Щелкните на кнопке *Мой компьютер* на панели задач, чтобы восстановить окно. Разверните окно на полный экран — щелкните на кнопке развертывания окна *Мой компьютер* (вторая справа в верхней строке окна — ☐), окно перейдет в полноэкранный режим. Кнопка развертывания превратится в кнопку восстановления (🗗). Щелкните на кнопке восстановления окна *Мой компьютер* для приведения его к нормальному виду.
- Закройте окно *Мой компьютер*. Для этого щелкните на кнопке закрытия окна или нажмите ALT+F4 (удерживая клавишу ALT, нажать клавишу F4).
- Откройте два окна программы *Мой компьютер* и попробуйте переключиться между ними с помощью мышки. Также для переключениями между окнами можно использовать комбинацию клавиш ALT+Tab.
- Проделайте аналогичные операции с вашей папкой *Программы*.

Запуск программ на выполнение

Прежде всего уточним, по каким внешним признакам можно отличить файл программы (так называемый исполняемый) от других типов файлов:

- во-первых, они имеют расширение «exe» или «com»;
- во-вторых, снабжены значком либо стандартного вида (изображение открытого окна с кнопками управления размерами), либо специальным.

Для запуска программ на выполнение существует два способа:

1) с помощью *Главного Меню*. Щелкните на кнопке *Пуск* на панели задач, а затем поищите нужную программу в пунктах меню (поиск осуществляется перемещением курсора по меню с непродолжительной задержкой на выбранных пунктах). Непосредственно запуск программы — щелчок левой кнопкой мыши на пункте меню с ее именем. Таким образом можно запустить программы, которые заранее были зарегистрированы в *Главном меню*;
2) с помощью программы *Мой компьютер* — поиском по дискам и папкам находится исполняемый файл программы и двойным нажатием левой кнопки мыши запускается на выполнение (универсальный способ для любых программ).

Этап 5

- Запустите на выполнение текстовый редактор *Блокнот* (NotePad). В соответствии с первым способом нажмите на кнопку *Пуск*, затем меню *Программы*, *Стандартные* и *Блокнот*. В соответствии со вторым способом в окне Мой компьютер откройте папку для диска, на котором размещена Windows (обычно это диск *C*:), откройте папку *WINDOWS*, найдите в ней значок *Notepad.exe* и активизируйте его, чтобы запустить программу. Обратите внимание, как изменилась панель задач при запуске программы.
- Запустите программу калькулятор (файл calc.exe в личном каталоге). Сколько и каких приложений запущено в данный момент в системе?
- Последовательно закройте все открытые окна — часть нажатием на кнопку закрытия в правом верхнем углу окна (☒), а часть — выбором соответствующей кнопки в панели задач, вызовом контекстного меню (правая кнопка мыши) и выбором команды

Закрыть. Можно воспользоваться сочетанием клавиш, универсальным для закрытия любого типа окон в пакетах Microsoft — ALT/F4.

Управление дисками, папками и файлами

Специальное приложение *Проводник* (рис. 41) предназначено для работы с файловой системой, предоставляя пользователю некоторые дополнительные возможности манипулирования, прежде всего с деревом папок (каталогов).

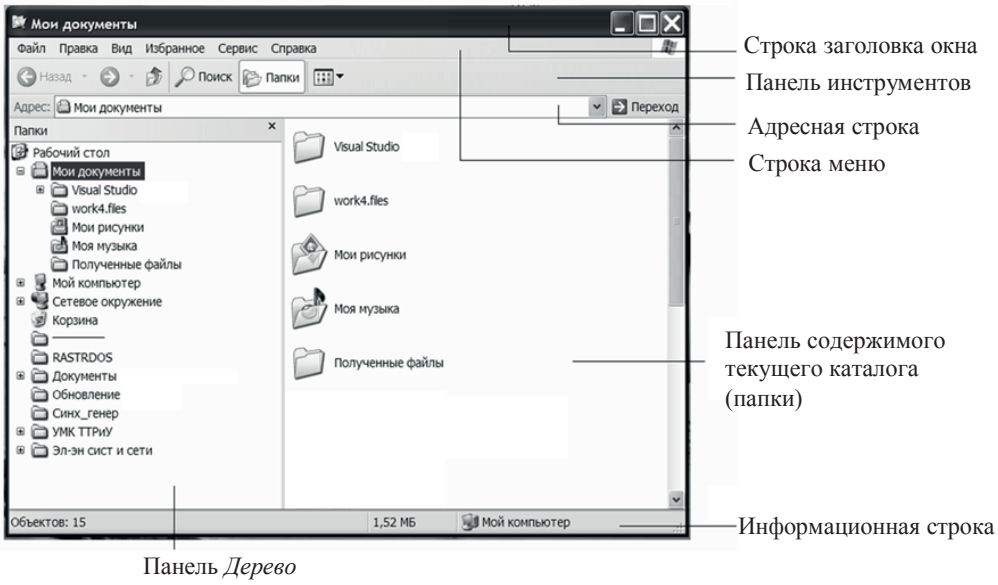


Рис. 43. Окно программы *Проводник*

Этап 6

- Запустите *Проводник*. Сделать это можно одним из трех способов:
  1. Откройте папку *Мой компьютер*. Выделите значок любого диска или папки и, удерживая клавишу *Shift*, щелкните дважды левой кнопкой мыши.
  2. Так же, как и в первом случае, выделите диск или папку и выберите из меню *Файл* команду *Проводник*.
  3. Третьим способом Проводник можно запустить из *Панели задач*: кнопка *Пуск*, меню *Все программы*, *Стандартные*, *Проводник*.

- Изучите внешний вид окна *Проводника*: меню, раскрывающийся список выбора, панель инструментов, панель дерева папок, панель содержимого папки. Щелкните на значке плюс (+) в панели дерева *Проводника*. Обратите внимание на изменения в панели *Папки*, а также в панели *Содержимое*...
- Скопируйте файл «r7.txt» из папки *P*:*LAR* в папку *Программы*. Если предварительно открыть папку, то сделать это можно обычным перетаскиванием с помощью правой (с активизацией контекстного меню режимов переноса) или левой кнопки мыши. В последнем случае, если перетаскивание происходит на другой диск или нажата при этом клавиша *Ctrl* клавиатуры, система выполняет копирование объекта. Можно предварительно не открывать папку, куда планируется копирование, а найти ее в панели *Папки*, если информация представлена в виде дерева *Проводника*. Искать, правда, следует осторожно: используя линейки прокрутки и значки «+» для демонстрации связей папки вниз по дереву. После этого остается только перетащить выбранный файл с помощью мыши на значок найденной папки. Обратите также внимание, куда предлагает *Проводник* скопировать файл в меню *Файл*, *Отправить*.
- Создайте папку с именем *Временная* с помощью *Проводника* в вашей папке *Программы*. Откройте для этого *Проводник*, щелкните на значке *Программы* и выделите его, чтобы в панели *Содержимое*... был представлен состав этой папки. Щелкните правой кнопкой на свободном пространстве панели содержимого. Из появившегося меню выберите пункт *Создать*, *Папка*. *Проводник* создаст новую папку. Наберите имя и нажмите *Enter*.
- Удалите созданную папку *Временная*, закройте *Проводник*. Удалить можно клавишей *Del* клавиатуры после того, как нужный объект выбран.

Поиск файлов и папок

Широкие возможности для поиска файлов и папок предоставляет пользователю программа *Найти*: *Файлы и папки*. Запустить ее можно либо из *Главного меню* (команда *Поиск*), либо из *Проводника* (меню *Сервис*, команда *Найти*). В окне программы с помощью соответствующей закладки задаются условия (или критерии) поиска.

Этап 7

- Найдите на диске *C*: все файлы, начинающиеся с букв *win*. Для выполнения задания можно поступить следующим образом: откройте *Проводник* и выберите значок диска *C*:. Выберите в меню *Сервис* команды *Найти*, *Файлы и папки*. На закладке *Имя* и размещение в текстовом окне *Имя* наберите *win** и щелкните по кнопке *Найти*. *Проводник* выполнит поиск на текущем диске и выведет в окно список всех файлов и папок, которые начинаются с букв *win*. Закройте *Проводник*.
- Найдите на диске *D*: файлы, которые были созданы или изменены за последние 10 дней. Для разнообразия запустим программу *Найти*: *Все файлы* из *Главного меню*. Нажмите на кнопку *Пуск* и выберите в меню команду *Поиск*, *Файлы и папки*. На закладке *Дата* изменения переключите радиокнопку в режим *Найти все файлы*, созданные либо измененные, включите радио-кнопку За последние ... дней, а затем задайте необходимый временной диапазон. Нажмите кнопку *Найти*.
- Повторите поиск с теми же критериями для остальных дисков компьютера, сделайте вывод об интенсивности использования дисков. Здесь, не изменяя параметров закладки *Дата изменения*, на закладке *Имя и размещение* в раскрывающемся списке выберите поочередно остальные диски.
- Найдите папку *Рабочий стол*. Сравните содержимое этой папки в окне программы *Мой компьютер* и в *Проводнике*. Какие можно сделать выводы?

Программа Корзина — восстановление удаленных объектов

Одной из системных папок Windows является папка Корзина, располагаемая на Рабочем столе, в которой некоторое время могут сохраняться объекты, удаленные пользователем. Запускается двойным нажатием левой клавиши мыши на значке Корзина на рабочем столе.

Этап 8

- Настройте режим удаления объектов с помещением их в корзину с подтверждением. Выделите значок *Корзина*, вызовите контекстное меню и выполните команду *Свойства*. На закладке *Глобаль-*

ные включите радио-кнопку Единые параметры для всех дисков и установите опцию (✓) Запрашивать подтверждение на удаление. Опция *Уничтожать файлы сразу после удаления*, не помещая их в корзину, должна быть снята!

- Удалите программу *Калькулятор* в папке *Программы*, а затем восстановите ее с помощью *Корзины*. Откройте для этого окно *Корзины*, найдите в нем файл *calc.exe*, выделите его, а затем в меню *Файлы* выполните команду *Восстановить*.

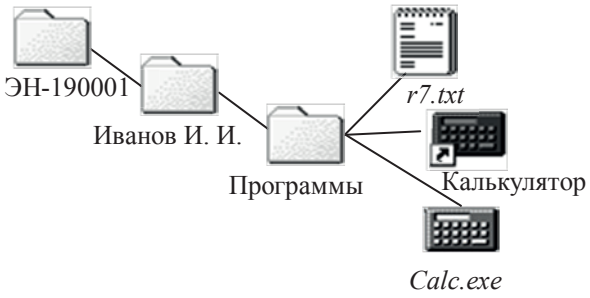


Рис. 42. Пример файловой системы

- Проверьте соответствие созданного вами фрагмента файловой системы, показанному на рис. 42.

Окончание работы

Кнопкой *Пуск* активизируйте *Главное меню*, выполните пункт *Завершение работы*, установите радио-кнопку в позицию *Выключить компьютер* и, дождавшись разрешительного сообщения на экране, выключите питание.

4. Основные приемы работы в редакторе Word

Word — это программа для создания или редактирования текстовых документов [2]. Помимо многочисленных возможностей представления текстовой информации, Word позволяет разнообразить текст рисунками, таблицами, выполнять сноски, структурировать документ, сформировать оглавление. Работа облегчается с использованием автокоррекции (трудно вводимые или длинные слова), автотекста (стандартные заготовки для документа), проверки орфографии и др.

Основные приемы управления в редакторе Word

Главный инструмент пользователя в редакторе Word — это мышь, а не клавиатура. Действительно, выбрать необходимые действия редактора из многочисленных средств управления и настройки, сгруппированных на рабочем столе, можно только с помощью «мышиного» курсора. На приведенном рисунке фрагмента рабочего стола редактора (рис. 43) указаны некоторые, но далеко не все доступные элементы. Самая первая — это строка названия окна редактора с кнопками управления окном (справа) и панелью быстрого доступа (слева).

В ранних версиях Word-а выбор одного из пунктов основного меню приводил к появлению «выпадающего» списка команд. В современных версиях меню организовано в виде «вкладок» и основные команды вынесены на «ленту» в виде полей. Например, на рис. 43 показаны поля, соответствующие вкладке или пункту меню *Главная*, а именно:

- поле *Буфер обмена*;
- поле *Шрифт*;
- поле *Абзац* и др.

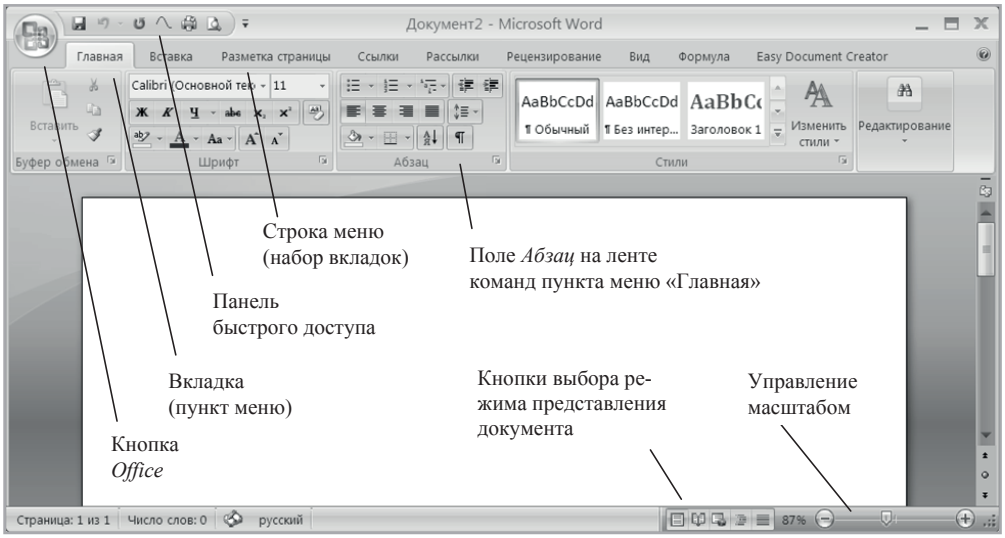


Рис. 43. Рабочий стол редактора

Каждое поле представляет собой набор инструментов, позволяющих выполнять действия или менять свойства текста/выделенного фрагмента.

Выбор первой кнопки, называемой кнопкой *Office*, позволяет выполнять действия со всем документом, а именно: создавать новый документ, открывать уже существующий, сохранять текущий (команда *Сохранить*), предусмотрена возможность сохранения копии документа в различных вариантах (команда *Сохранить как*), печать документа и т. д. Кроме того, в поле *Параметры Word* устанавливаются некоторые настройки характеристик документа: отображения на экране, правописания, параметров автосохранения и др., которые являются общими для текущего документа.

Вертикальной пульсирующей чертой (курсор ввода) обозначена активная позиция рабочего поля.

В нижней строке окна документа находятся кнопки выбора режима представления документа:

- кнопка обычного режима — текст показывается сплошной «лентой», без разрывов на страницы. Данный режим удобен для первой стадии подготовки документа — набора текста;
- кнопка режима электронного документа. Данный режим позволяет сделать текст и фон цветными, использовать простейшие анимационные эффекты и пр.;

- кнопка режима разметки страниц — текст показывается в таком же виде, в каком он будет впоследствии (возможно) напечатан;
- кнопка режима структуры документа. Данный режим используется для подготовки достаточно серьезных текстов с использованием заголовков и подзаголовков различных уровней.

Справа от кнопок располагается горизонтальная полоса прокрутки — для быстрого изменения масштаба представления. В этой же строке расположены текущие характеристики документа и режима работы.

Этап 1

- Запустите редактор. Ознакомьтесь с элементами рабочего стола. Найдите указанные на рисунке элементы на экране. Сравните, чем отличаются набор элементов управления на рисунке и на вашем компьютере. Устраните различия, если они есть.
- Откройте пустое окно документа — кнопка *Office*, команда *Создать* и из предложенного списка выберите вариант — *Пустые и последние*, *Новый документ*.
- Установите предназначение трех-пяти кнопок на панелях инструментов. Для этого курсор мыши наводится на интересующую вас кнопку и через небольшую паузу появляется окно-подсказка. В случае, если подсказки нет, зайдите через кнопку *Office* в область *Параметры Word*, найдите область *Экран* и в поле *Параметры отображения страниц* отметьте опцию *Показывать всплывающие подсказки при наведении указателя мыши*.
- Найдите на диске *P*: файл *d1.doc* и загрузите его в окно документа (если требуемый файл найти не удается, обратитесь к преподавателю). Поиск и загрузка файла выполняется через кнопку *Office* с помощью команды *Открыть*. Если неизвестно местонахождение файла, воспользуйтесь стандартными средствами поиска операционной системы, рассмотренными в предыдущих работах.
- Установите режим просмотра документа *разметка страницы*. Подсчитайте, сколько абзацев содержит документ. Установите режим просмотра документа с изображением спецсимволов *конец абзаца*. Сколько таких спецсимволов расставлено по тексту? Для этого найдите через кнопку *Office* и *Параметры Word* область *Экран*, в поле *Всегда показывать эти знаки форматирования на экране* отметьте *Знаки абзацев* ¶.

- Добавьте в конец текста новый абзац следующего содержания: «Основная идея системы Windows и программ, составляющих пакет Microsoft Office — «дружественный» или интуитивный интерфейс (т. е. максимально возможное удобство в работе)».
  Спецсимволы абзаца будут появляться в тексте всякий раз, как вы нажмете на клавишу *Enter*. Отсюда вывод — не нужно выполнять нажатие *Enter* для перехода на новую строку — Word сам это сделает за вас.
- Выделите слово «Windows» в первом абзаце и смените его начертание на жирный курсив. Первый способ — мышью наводится курсор на данное слово в тексте, два раза нажимается левая клавиша, а затем нажимаются кнопки Ж и К в поле *Шрифт* вкладки *Главная*. Либо выделяется слово и на появляющейся справа всплывающей панели форматирования нажимаются клавиши Ж и К. И третий способ — после выделения слова надо нажать правую клавишу мыши, в появившемся контекстном меню выбрать команду *Шрифт* и в поле *Начертание* установить *Полужирный курсив*.
- Измените параметры первого абзаца текста с помощью линейки (левая граница, правая граница, абзацный отступ). Если линейка отсутствует на рабочем столе, воспользуйтесь вкладкой *Вид*, в поле *Показать или скрыть* отметьте *Линейку*. Затем установите курсор на любом слове абзаца. После чего наведите курсор мыши на один из ограничителей шкалы (их на линейке три), нажмите левую клавишу мыши и «перетащите» выбранный ограничитель (не отпуская клавиши) на новое место (прием называется «взял и тащи»).
- Сделайте все абзацы текста по образцу первого абзаца. Работа с линейкой аналогична предшествующему пункту, но предварительно следует выделить весь текст, например с помощью комбинации Ctrl+A.
  Для выделения части текста существуют несколько способов: мышью — наводится курсор на данное слово в тесте, нажимается левая кнопка мыши и мышь протаскивается по слову с удерживаемой левой кнопкой. Клавиатурой: курсор ввода устанавливается перед выделяемым фрагментом, нажимается клавиша *Shift* клавиатуры и, не отпуская ее, клавишами-стрелками область выделения растягивается на весь фрагмент.

- Замените начертание слова Windows во всем тексте на подчеркнутое. Найдите на вкладке *Главная* поле *Редактирование*. Выберите команду *Заменить* и вам откроется диалоговое окно замен. Если вызывать режим замены при уже выделенном слове, Word в поле для *Найти* уже сам поместит указанное слово. Также можно воспользоваться еще одним средством — буфером обмена.

Для этого, во-первых, если это необходимо, «отбуксируйте» диалоговое окно замен по рабочему столу так, чтобы открылся кусочек редактируемого текста со словом «Windows» (ведь именно его мы собираемся заменять!). Буксировка — это тоже «взял и тащи», но зацепить нужно заголовок диалогового окна. Затем выделите слово «Windows» в тексте и выполните Ctrl+Ins (или Ctrl+C) — тем самым вы выполнили операцию *копировать в буфер обмена* (буфер обмена — это временное хранилище информации). Вернитесь в диалоговое окно *Заменить* (просто щелкните на нем), установите курсор в поле *Найти* и выполните *Shift+Ins* для вставки фрагмента из буфера обмена (или Ctrl+V). Таким же образом вставьте слово «Windows» и в поле *Заменить* на, а затем, не убирая курсор из данного поля, нажмите кнопку Ч на панели инструментов Форматирование. Нажмите *Заменить все*, проверьте, произошли ли необходимые замены. В том случае, если была произведена только одна замена, установите параметры замены, то есть находясь в поле замены (слово можно выделить), нажмите клавишу *Больше*, выберите в выпадающем списке из *Формат* закладку *Шрифт* и установите необходимый формат для заменяемого слова. Выполнив команду *Заменить все*, сравните с предыдущим результатом.

Кроме сочетаний клавиш, упомянутых выше по тексту, в Microsoft Word назначено по умолчанию большое количество других сочетаний клавиш для более простого и быстрого запуска операций, которые и называются поэтому быстрыми клавиши. Наиболее используемые:

Быстрые клавиши	Выполняемые функции
Ctrl + S	Сохранить документ
Ctrl + Z	Отменить выполненное действие
Ctrl + Y	Вернуть отмененное действие
Ctrl + X	Вырезать выделенный текст в буфер обмена
Ctrl + B	Применить полужирное начертание к выделенному тексту
Ctrl + I	Применить курсивное начертание к выделенному тексту

Быстрые клавиши	Выполняемые функции
Ctrl + U	Подчеркнуть выделенный текст
Ctrl + L (R)	Выровнять содержимое текста по левому (правому) краю
Ctrl + E	Выровнять содержимое текста по центру
Ctrl + J	Выровнять содержимое текста по ширине
Ctrl + F	Открыть поле навигации, с помощью которого можно быстро перемещаться по конкретным главам, вывести миниатюры страниц, найти конкретный текст
Ctrl+Shift+Space	Вставить неразрывный пробел (пробел, по которому не выполняется перенос на следующую строку)

Примечание. Если требуется выделить большой фрагмент текста (например, несколько страниц), то, вместо долгого выделения мышкой (с нажатой левой клавишей), можно поставить курсор в начало выделяемого фрагмента, пролистать документ на конец выделяемого фрагмента и, при нажатой клавиши *Shift,* указать курсором мышки нужное место. Можно отметить, что в общем случае клавишей *Shift* выполняется выделение смежных, а клавишей *Ct*rl — произвольно расположенных объектов.

- Проверьте комбинации быстрых клавиш в работе.
- Сохраните полученный текст в вашей личной папке с именем *fio.doc*, где *fio* — это ваши инициалы. Для этого в меню *Файл* выполните команду *Сохранить как*, указав новое имя в поле *Файл* и установив тип файла «Документ Word».

Этап 2

- Откройте файл, который вы сохранили в конце выполнения Задания 1.
- Добавьте в него четыре следующих предложения в качестве заголовков абзацев (каждое помещается перед очередным абзацем текста и заканчивается, в свою очередь, символом абзац — «¶»):
  Новые средства Windows;
  GUI — графический пользовательский интерфейс;
  Мультизадачный режим;
  ИНФОРМАЦИОННЫЙ обмен.
  Примечание. Слово «Информационный» по замыслу должно быть записано прописными (заглавными) буквами. Рекомендуется ввести его сначала строчными (малыми) буквами, затем выделить (например, двойным кликом) и, находясь указателем мышки на выделении, вызвать правой клавишей мышки контекстное

меню, а в нем команду *Шрифт*. В разделе *Видоизменение* открывшегося при этом окна с помощью соответствующего чекбокса (*check box* — окошко выбора) установите режим прописного написания данного слова. Проверьте работу остальных вариантов выбора.

- Полезно при подготовке различного рода документов автоматизировать процесс формирования оглавления. Для этого встаньте курсором на первую строку, на вкладке *Ссылки* выполните команду *Оглавление* и, в открывшемся списке, выберите *Автособираемое оглавление 1.* В результате в тексте появится строка *Оглавление* и сообщение о том, что элементы оглавления не найдены. Причина в том, что пока в документе заголовки не оформлены соответствующим образом. Оформление сводится к выбору для каждого заголовка одного из стилей: Заголовок 1, Заголовок 2, Заголовок 3, Подзаголовок. Список всех стилей можно увидеть на вкладке *Главная* в поле *Стили*. Примечательно, что стили могут быть выбраны не только для элементов оглавления, но и для других фрагментов текста. По умолчанию всему тексту присваивается стиль *Обычный*. Чтобы выбранный вами текст имел определенный стиль, необходимо выделить данный текст, а затем нажать кнопку выбранного вами стиля. Если стиль необходимо применить только к одному абзацу, то выделять его необязательно: достаточно просто поставить курсор на данный абзац и выбрать стиль. Для ранее введенных оглавлений абзацев выберите стиль *Заголовок 1.* Выполните команду *Обновить поле* в режиме *Обновить целиком* в контекстном меню оглавления (находясь на строке с сообщением о ненайденных элементах).
- Измените формулировку одного или нескольких заголовков документа и выполните обновление оглавления. Продемонстрируйте результат преподавателю.

  Редактор предоставляет широкие возможности по изменению стилей. Для этого в поле *Стили* выбирается стиль, и в его контекстном меню (клик на правой кнопке мыши) выполняется команда *Изменить*. В появившемся окне в поле собраны элементы для изменения основных параметров стиля. Если нажать на кнопку *Формат*, то появится список с дополнительными параметрами. Установите характеристики стилей для редактируемого документа в соответствии с табл. 10.

Таблица 10

Стиль текста	Обычный	Заголовок 1	Заголовок 2
Шрифт	Times New Roman	Courier	Arial
Размер шрифта	12	16	14
Начертание	Обычный	Полужирный курсив	Курсив
Междустрочный интервал	Одинарный	Полуторный	Полуторный
Выравнивание текста	По ширине	По центру	По правому краю
Отступ первой строки	1,25	0	0

Примечание. Изменить стиль фрагмента текста, используя параметры какого-либо другого фрагмента, можно с помощью кнопки *Формат по образцу,* которая расположена на вкладке *Главная* в зоне *Буфер обмена*.

- Создайте в конце текста таблицу по образцу табл. 11. Построение таблицы начните с нажатия кнопки *Таблица* на вкладке *Вставка*. В открывшемся окошке на поле из квадратиков мышью выбирается размер таблицы (в данном случае — 13 строк и 2 столбца). Редактор, как правило, ограничивает размер заготовки таблицы. В этом случае необходимо выбрать максимально возможный размер (например, 2 на 8), а затем добавить необходимое количество строк или столбцов в уже созданную таблицу. Предположим, в нашем случае не хватает пяти строк. Для их добавления выделите пять любых строк таблицы и, находясь на них курсором, в контекстном меню выполните команду *Вставить строки*. В первой строке образца объединенная ячейка. Чтобы соответствовать этому, выделите первую строку и в контекстном меню выберите команду *Объединить ячейки*. На этом структура таблицы подготовлена, можно вводить текст. Перемещаться между ячейками строки удобно клавишами — стрелками или клавишей Tab.

Таблица 11

Поколения системы Windows

№ п/п	Название версии	Год появления
1	Windows 1.0	1985
2	Windows 2.0	1987
3	Windows 95	1995
4	Windows 98	1998
5	Windows ME	2000

Окончание табл. 11

№ п/п	Название версии	Год появления
6	Windows XP	2001
7	Windows Vista	2007
8	Windows 7	2009
9	Windows 8	2012
10	Windows 10	2015

Примечание. С помощью вкладок *Конструктор* и *Макет* можно значительно улучшить внешний вид полученной таблицы. Познакомьтесь с этими возможностями.

- Вставьте в таблицу слева дополнительный столбец. При необходимости выровняйте ширину столбцов, чтобы таблица не выходила за границу листа. Еще раз выполните объединение для верхней ячейки таблицы (табл. 12).
- В новом столбце планируется разместить порядковые номера записей. В верхней ячейке наберите на клавиатуре «№». После этого выделите оставшиеся пустые ячейки столбца и нажмите кнопку *Нумерация* на вкладке *Главная* в поле *Абзац*.
- Оформите выравнивание столбца «№», чтобы новые номера находились в пределах видимости таблицы. Для этого необходимо нажать на ячейку с номером таблицы и на верхней линейке отрегулировать выравнивание при помощи флажков, отвечающих за отступ текста. Если данные флажки не отобразились, то необходимо временно увеличить ширину столбца. После оформления выравнивания верните столбцу прежнюю ширину.

Таблица 12

Поколения системы Windows

№	Название версии	Год появления
1	Windows 1.0	1985
2	Windows 2.0	1987
3	Windows 95	1995
4	Windows 98	1998
5	Windows ME	2000
6	Windows XP	2001
7	Windows Vista	2007
8	Windows 7	2009
9	Windows 8	2012
10	Windows 10	2015

- Добейтесь, чтобы ширина столбцов «Название версии» и «Год появления на рынке» были одинаковыми. Для этого необходимо выделить ячейки выравниваемых столбцов и на появившейся вкладке *Макет* в поле *Размер ячейки* нажать кнопку *Выровнять ширину столбцов*. Результат работы продемонстрируйте преподавателю.
- Удалите две верхние строчки таблицы, а оставшуюся часть отсортируйте по годам появления Windows на рынке, начиная с самой ранней версии (Windows 10). Для этого нажмите на любую ячейку таблицы и на появившейся вкладке *Макет* в поле *Данные* нажмите кнопку *Сортировка*. В появившемся окне (рис. 44) необходимо выбрать сортировку по *столбцам 3*, тип *Число* или *Дата*, *по убыванию* и нажать Ок.

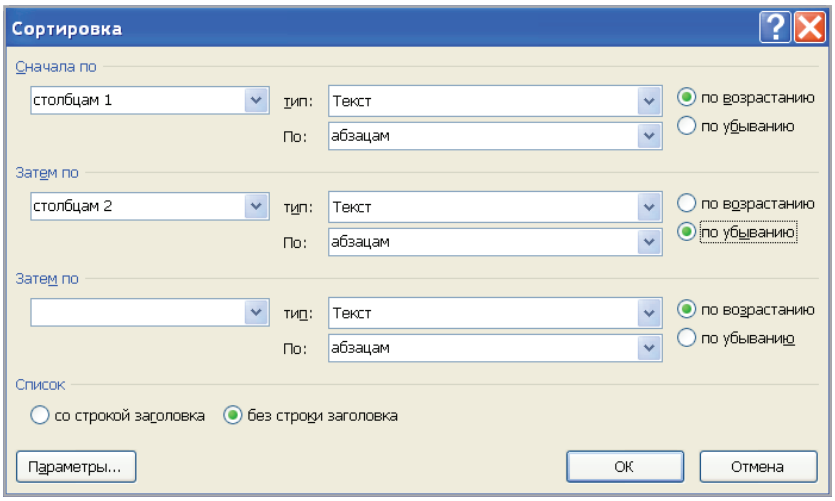


Рис. 44. Окно *Сортировка*

- Проверьте действие иных методов сортировки.
- Добавьте нумерацию страниц в документе. Для этого на вкладке *Вставка* в поле *Колонтитулы* нажмите кнопку *Номер страницы*, а затем выберите из списка стиль установки номеров страниц. Закройте окно *Колонтитулы* и продемонстрируйте результат работы преподавателю. Сохраните документ на диске в вашей личной папке.

5. Основы работы в Excel

Приложение Excel в настоящее время является самым популярным табличным процессором [2, 3]. Его возможности позволяют эффективно решать широкий диапазон задач, начиная от работы с обычными таблицами и заканчивая операциями со средних размеров базами данных. Настоящее занятие позволит освоить основные приемы работы с этой программой.

Рабочий стол Excel

Документ Excel называется «рабочая книга» и состоит из нескольких листов. Лист — это таблица прямоугольных ячеек, каждая из которых имеет свой адрес, который формируется по примеру «шахматной доски». Имеется два формата адресов:

- *БуквоЦифра*, где Буква (большая, латинского алфавита) — это обозначение столбца, а Цифра — номер строки;
- *RномерстрокиLномерстолбца*, где номер строки и номер столбца являются обычными координатами.

«Координатные оси» и в первом и во втором случае расположены по верху и левому краю листа. Наиболее наглядны адреса в формате «БуквоЦифра», поэтому он и используется чаще всего. Адреса, записанные в соответствии с данными форматами, называются *относительными*. Данное свойство проявляется в случае перемещения по листу формул, в которые они могут входить. Если адрес (в формуле) необходимо жестко «привязать» к некоторой ячейке, используют так называемые *абсолютные* адреса. От относительных они отличаются тем, что зафиксированная координата начинается с символа $ (например, A1 — абсолютный адрес верхней левой ячейки рабочего листа, ко-

торая становится активной при открытии новой книги или нового листа в книге). Если необходимо задать адрес ячейки, которая находится на другом листе, перед адресом указывается имя листа. Например, запись «Лист2! С6» определяет на втором листе ячейку в столбце «С» и в строке «6». Знак «!» является здесь разделителем.

Любые данные в Excel должны принадлежать какой-либо ячейке (рис. 45). Ячейка, содержимое которой в данный момент может быть изменено пользователем, называется активной и выделяется на листе жирной рамкой. Координаты активной ячейки (А1 на рис. 45) воспроизводятся в специальном окне-списке в левой верхней части рабочего стола (в строке под панелями инструментов), а ее содержимое дублируется правее в «строке формул».

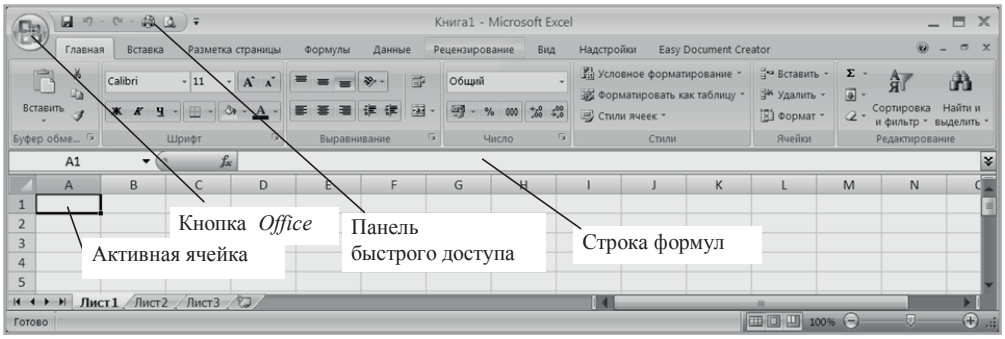


Рис. 45. Рабочий стол *Excel*

Существует два способа изменения данных в ячейке.

1. Начальный ввод или полная замена. После того, как ячейка сделана активной, осуществляется набор необходимых данных на клавиатуре. Окончание ввода — клавиши управления курсором (стрелки), табуляции или *Enter*. Если вместо указанных клавиш нажимается клавиша *Esc*, ввод отменяется и в ячейке остается прежнее значение.

2. Редактирование значения ячейки. На нужной ячейке выполняется двойное нажатие левой кнопки мыши (рамка вокруг ячейки изменяет свой вид), и в месте, обозначенном курсором текстового ввода (пульсирующая вертикальная черта), осуществляется ввод новых или удаление существующих символов. Здесь клавиши-стрелки выполняют перемещение курсора внутри редактируемой ячейки. Окончание ввода — *Enter* или щелчок мыши на другой ячейке, отмена — клавиша *Esc*.

Данный режим носит название «Редактирование прямо в ячейке». Если он не установлен, существует возможность редактирования содержимого ячейки в строке формул. Для этого ячейка делается активной (стрелками или щелчком на левой кнопке мыши), а затем также однократное нажатие левой кнопки мыши выполняется в любом месте строки формул. Установить режим редактирования в ячейке можно через кнопку *Office*, *Параметры Excel*, команда *Дополнительно* (определяет дополнительные параметры для работы с Excel), поле *Параметры правки*, поставив галочку для *Разрешить редактировать в ячейках*.

Разлиновка листа на ячейки — служебная и на печать не выводится.

Для обрамления необходимых ячеек (например, при оформлении таблицы) пользуются специальными кнопками на закладке *Главная*, поле *Шрифт* (рис. 46).

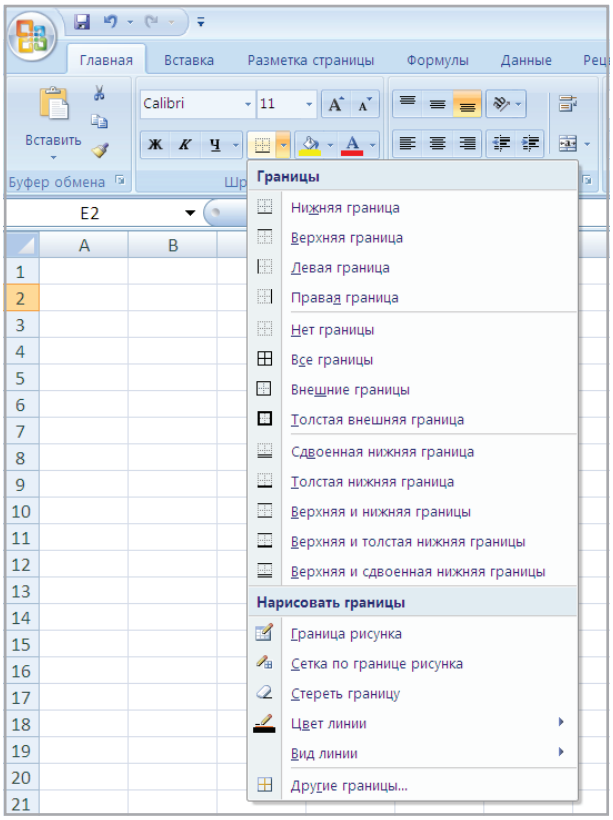


Рис. 46. Задание обрамления ячейки

Некоторые проблемы могут быть при вводе числовых данных, представляющих собой числа с дробной частью. Стандартными установками операционной системы Windows предусматривается в качестве разделителя дробной и целой части использовать запятую. Если это непривычно, необходимо изменить установки в *Дополнительных параметрах* работы в *Excel* в области *Параметров правки* снять галочку *Использовать системные разделители* и поставить свой *Разделитель целой и дробной части*. Если этого не сделать, то когда число вводится с другим разделителем, Excel воспринимает данные как текст.

Этап 1. Подготовка ведомости учета оплаты электроэнергии

Представим себе некоторое предприятие, осуществляющее поставку электроэнергии конечным потребителям. Для упрощения положим, что электроэнергия отпускается физическим лицам и по одному фиксированному тарифу (допустим, 40 копеек за кВт·час).

Приложение Excel должно быть запущено на вашем компьютере. Работу начнем на первом рабочем листе.

Заготовка таблицы

Назовем таблицу «Оплата счетов за электроэнергию». Для этого в ячейку B1, предварительно сделав ее активной, с помощью клавиатуры вводим упомянутую фразу. Чтобы в названии таблицы использовался шрифт *Times New Roman* размером 12 пунктов, выберите соответствующие параметры с помощью списков поля *Шрифт* на закладке *Главная*.

Примечание. Пункт — это типографская величина измерения размера букв, равная 1/72 части дюйма.

Следующим шагом подготовим шапку таблицы. В ней должны быть столбцы: «№», «Потребитель», «Месячный расход», «Сумма счета», «Дата счета», «Сумма оплаты», «Дата оплаты». Название каждого столбца вводим в соответствующую ячейку второй строки, начиная со столбца A. Результат должен соответствовать рис. 47.

Тем не менее, удовлетворительным его назвать пока нельзя. Сначала установим на место заголовок. Для этого выделим ячейки диапазона A1:G1 либо мышью (щелкнув на A1 и, не отпуская левой кнопки, протащив манипулятор до ячейки G1), либо стрелками (переместиться в A1 и, удерживая *Shift*, стрелкой → расширить диапазон выделе-

ния). Затем для выделенного диапазона выполняется команда *Объединить* и поместить в центре панели команд *Выравнивание* (рис. 47).

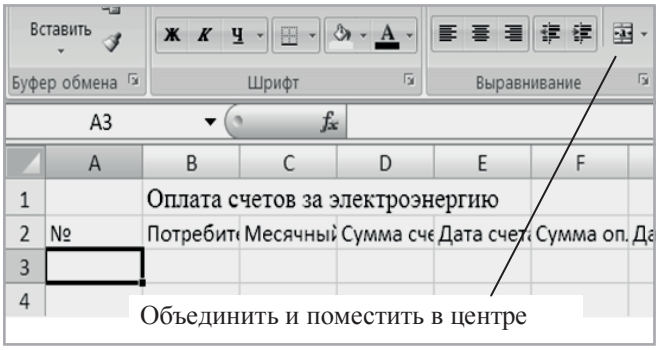


Рис. 47. Первый этап создания таблицы

Далее изменим формат «шапки». Ширину столбца А откорректируем вручную. Для этого курсор мыши наведем на линию, разделяющую буквы А и В на горизонтальной «координатной оси» (курсор превратился в стрелку), и, зацепив эту линию левой кнопкой мыши, перетащим несколько левее для уменьшения ширины столбца. Подобным образом могут быть откорректированы и строки (если требуется). Остальные столбцы отформатируем совместно. Для этого выделим диапазон A2:G2. Затем, оставаясь мышью в границах выделенного диапазона, нажмем правую кнопку мыши для вызова контекстного меню. Состав контекстное меню зависит от типа объекта, для которого оно открывается и содержит самые используемые команды. В данном случае нам нужна команда *Формат ячеек*. На ее закладке *Выравнивание* установите параметры по образцу рис. 48. При необходимости несколько увеличьте ширину столбцов В и С (вручную) и выполните обрамление на 10 строчек (без учета шапки). Результат вашей работы должен соответствовать рис. 49.

Заполнение таблицы

Сначала пронумеруем строки заполняемой таблицы. Первые две строки — вручную. В ячейку А3 вводим единицу, нажимаем *Enter* и оказываемся там, где и нужно — в ячейке А4. Если этого не произошло, проверьте в установках параметров Excel (рис. 50). В ячейку А4 введите число 2. Остальные номера ячеек заполним с использованием *Автозаполнения*. Для этого мышью или клавиатурой выделяем диапазон

A3:A4 и наводим курсор мыши на правый нижний угол этого диапазона. Курсор мыши должен превратиться в крестик (рис. 51). Затем, нажав и не отпуская левую кнопку мыши, нужно «растянуть» первоначально выделенный диапазон на все ячейки столбца. В результате Excel сам расставит номера строк в возрастающем порядке.

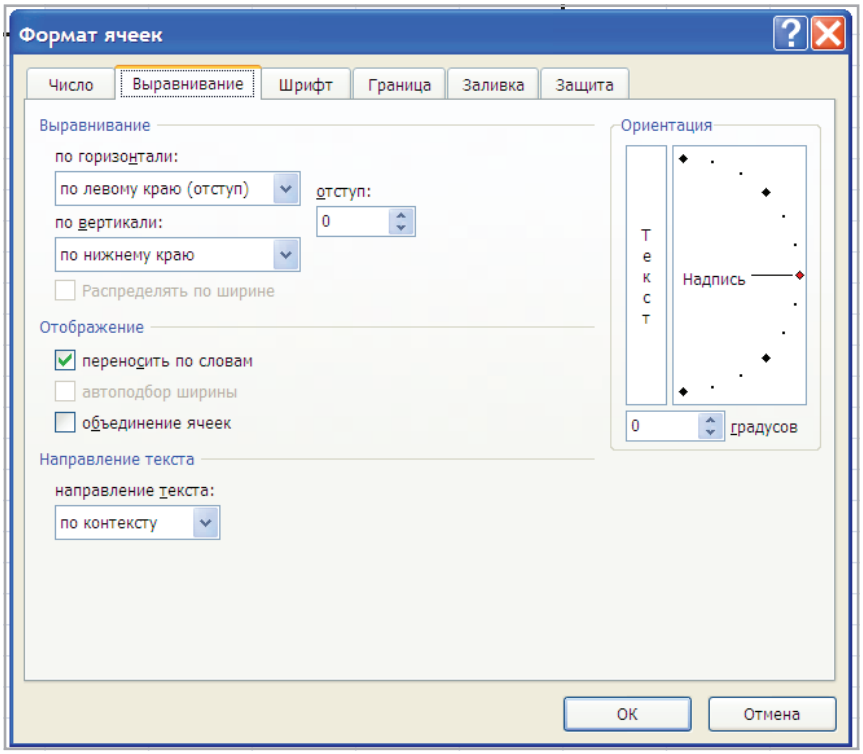


Рис. 48. Диалог Формат ячеек

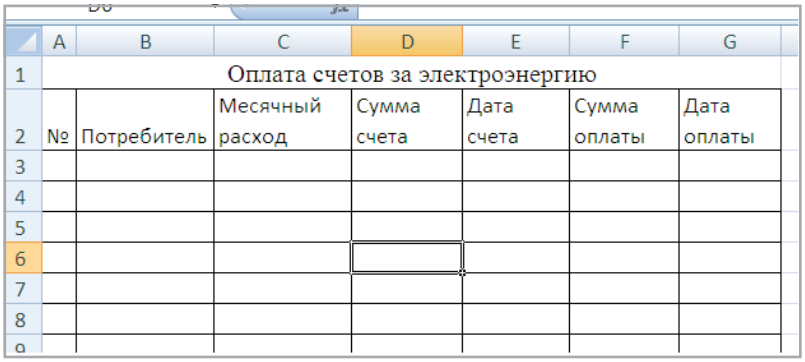


Рис. 49. Элементы шапки таблицы

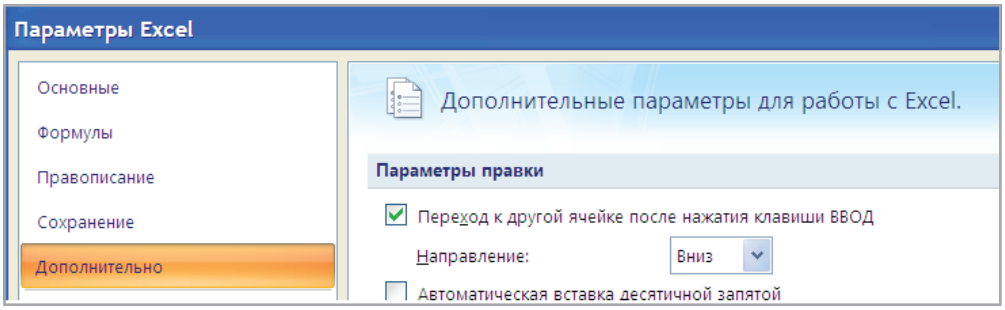


Рис. 50. Установка дополнительных параметров

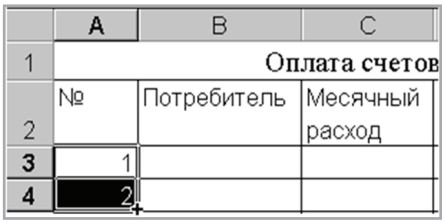


Рис. 51. Режим автозаполнения

Для автозаполнения второго столбца таблицы предварительно определим свой список. В порядке упрощения будем считать, что имеется только три потребителя: Иванов, Петров Сидоров. Данные в таблицу будем заносить помесячно, поэтому в столбце «Потребитель» указанная последовательность фамилий будет в трех экземплярах.

В данном случае можно воспользоваться услугами *Автозаполнения*, как было сделано для предыдущего столбца (т. е. можно набрать в первых трех ячейках столбца «Потребитель» указанные фамилии, выделить ячейки и «растянуть» их до конца столбца), но можно использовать и другой способ. Воспользуемся возможностью Excel создавать список. Для этого выполните следующую последовательность действий: нажав кнопку *Office*, вызовите *Параметры Excel*, среди установок *Основные* (в группе *Основные параметры при работе с Excel)* нажмите *Изменить списки*. Далее напротив выделения *Новый список* (рис. 52) в поле *Элементы списка* введите фамилии (через запятую). Для завершения задания нового списка нажимаем на кнопку Добавить и кнопку OK. Остается использовать новый список в деле. В первую ячейку второго столбца нашей таблицы вручную записываем фамилию Иванов, а затем растягиваем диапазон, состоящий в данном случае из одной ячейки, на весь столбец.

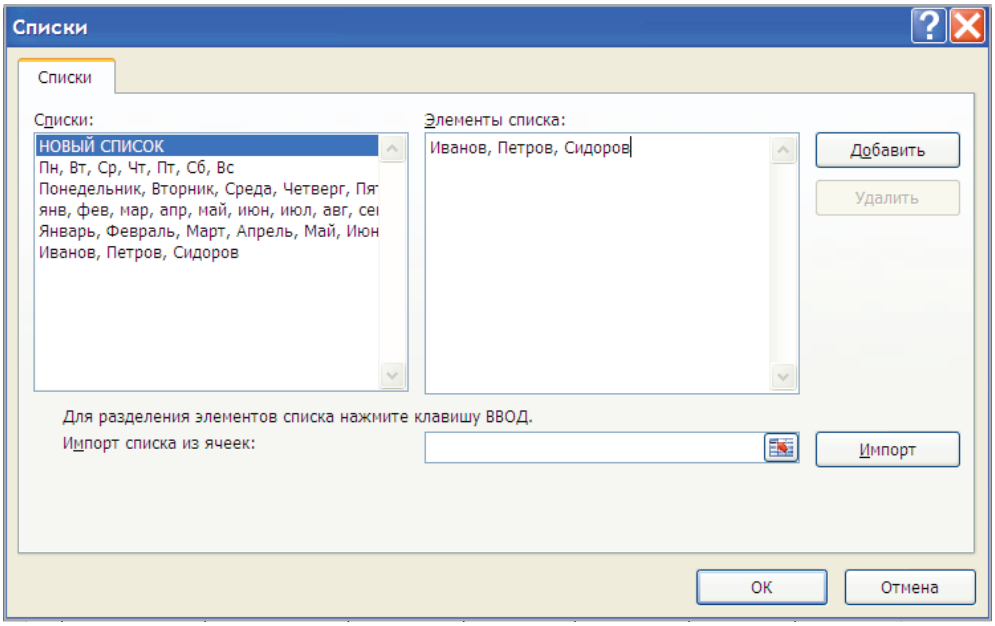


Рис. 52. Создание нового списка

Следующий на очереди — столбец расхода электроэнергии в месяц. Для простоты предположим, что потребление электроэнергии увеличивается от потребителя к потребителю и из месяца в месяц, начиная с 10, и на 10 кВт часов ежемесячно. Заполните столбец «Месячный расход», также используя *Автозаполнение*. Последнюю строчку таблицы, в которой позже будут подводиться общие итоги, пока необходимо очистить (например, клавишей *Del*).

Далее необходимо заполнить столбец «Сумма счета», в котором следует рассчитать стоимость электроэнергии в зависимости от расхода за месяц. Для этого в ячейку D3 необходимо поместить формулу. Запись формулы начинается знаком «=». После этого записывается расчетное выражение, в котором адреса необходимых ячеек связываются знаками арифметических операций. В данном случае можно ввести в ячейку D3 следующее: = C3 * 0,4. Правильнее не вводить адрес ячейки вручную, так как латинская «C» и русская «С» имеют одинаковое начертание, но воспринимаются Excel как разные символы. Удобнее и быстрее то же самое выполнить по-другому.

	A	B	C	D
1			Оплата счетов за э	
2	№	Потребитель	Месячный расход	Сумма счета
3	1	Иванов	10	
4	2	Петров	20	
5	3	Сидоров	30	
6	4	Иванов	40	
7	5	Петров	50	
8	6	Сидоров	60	
9	7	Иванов	70	
10	8	Петров	80	
11	9	Сидоров	90	

Рис. 53. Промежуточное состояние таблицы

Дело в том, что после знака «=» Excel позволяет явно указать операнды выражения, наводя курсор на нужные ячейки, либо мышью, либо клавишами — стрелками клавиатуры. Непосредственно с клавиатуры остается ввести только знаки операций и значение констант, если таковые имеются в формуле. В итоге последовательность задания формулы в ячейке должна быть следующей: активировать ячейку D3, нажать на клавишу «=», подвести указатель мыши к той ячейке, содержимое которой будет использовано в формуле, в нашем случае это C3, и один раз нажать левую клавишу мыши. Сразу в ячейке D3 появится адрес C3, причем выделенный цветом, курсор ввода встанет после адреса и дополнительно мигающим пунктиром останется выделенной ячейка C3. Далее на клавиатуре набираем «*0,4» и, как только ввод формулы завершен (нажата *Enter*), Excel выполняет по ней вычисления, а результат помещает в данную ячейку.

Формулу, введенную в ячейку D3, можно использовать как образец для автозаполнения остальных ячеек данного столбца. Для этого нужно «растянуть» диапазон, состоящий из ячейки D3, на весь столбец. Предварительный анализ значений, появившихся в столбце, позволяет надеяться, что Excel сам разобрался и «подправил» формулы в процессе автозаполнения. Убедиться, что это действительно так, можно просмотрев формулы в столбце D по очереди (рис. 53). Можно все формулы на листе увидеть сразу, выбрав команду *показать формулы*, относящуюся к группе *Зависимости формул*, которая, в свою очередь, расположена на вкладке *Формулы* (рис. 54).

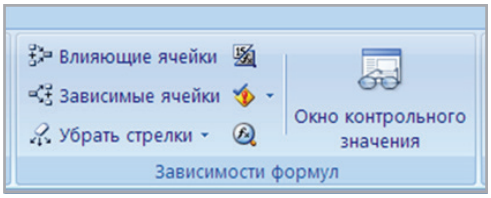


Рис. 54. Группа Зависимости формул

Убедившись, что все пока верно, нажмите кнопку *показать формулы* еще раз и посмотрите, какие дополнительные возможности представления формул предлагает Excel. Чтобы появилась всплывающая подсказка, достаточно подвести к интересующей команде указатель мыши. Если этого не происходит, надо выбрать через кнопки *Office*, *Основные Параметры Excel* и *Стиль всплывающих подсказок* опцию *Показывать улучшенные всплывающие подсказки*.

Следующий на очереди столбец «Дата счета». Предположим, что таблица формируется для первого квартала и счет выставляется в последний день месяца.

Целесообразно сначала, выделив весь столбец E, с помощью команды *Формат Ячеек* контекстного меню установить на закладке *Число* параметры в соответствии с рис. 55.

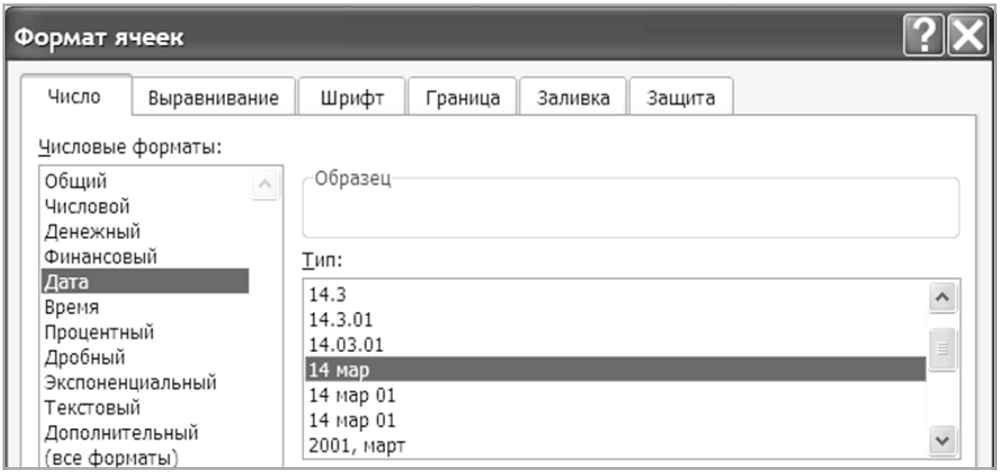


Рис. 55. Настройка формата даты

После этого можно ввести первую дату «31.01.». Ее значение тут же будет преобразовано в заданный формат и показано в ячейке. В качестве

образца для автозаполнения ячейка с датой использована быть не может (Excel начнет увеличивать дату). В данном случае можно воспользоваться средствами копирования и восстановления через *Буфер обмена*. Для этого, оставаясь в ячейке-источнике (откуда копировать), выбирается команда *Копировать* из группы *Буфер обмена* на вкладке *Главная* (или Ctrl+Ins), а затем, переместившись в ячейку-приемник (куда копировать), выполняется команда Вставить из той же группы команд (или Shift+Ins). Подобным образом заполняем столбец до конца.

Excel предоставляет возможность создавать собственные форматы ячеек. Для этого в окне *Формат Ячейки* необходимо выбрать *Все форматы* и в поле *Тип* ввести маску необходимого формата. Символы, которые при этом можно использовать, приведены в табл. 13.

Таблица 13

№	Спецсимвол	Назначение спецсимвола
1	0 (нуль)	Обязательный разряд
2	# (решетка)	Необязательный разряд
3	(пробел)	Разделитель групп разрядов
4	[] (квадратные скобки)	Внутри скобок записывается цвет текста из следующего списка: белый, черный, синий, красный, желтый, голубой, зеленый

Если имеется необходимость в том, чтобы выводились конкретные единицы измерения (кВт·ч, кВт, А, В и т. д.), необходимо задать их при формировании маски, используя кавычки.

	A	B	C	D	E	F	G
1	Оплата счетов за электроэнергию						
2	№	Потребитель	Месячный расход	Сумма счета	Дата счета	Сумма оплаты	Дата оплаты
3	1	Иванов	10	4	31 янв	3,2	10 фев
4	2	Петров	20	8	31 янв	6,4	10 фев
5	3	Сидоров	30	12	31 янв	9,6	10 фев
6	4	Иванов	40	16	28 фев	12,8	10 мар
7	5	Петров	50	20	28 фев	16	10 мар
8	6	Сидоров	60	24	28 фев	19,2	10 мар
9	7	Иванов	70	28	31 мар	22,4	10 апр
10	8	Петров	80	32	31 мар	25,6	10 апр
11	9	Сидоров	90	36	31 мар	28,8	10 апр
12							

Рис. 56. Подготовленная таблица

Следующий на очереди столбец — «Сумма оплаты» (по ранее выставленному счету). В реальных условиях это, конечно, внешние данные и вводить их пришлось бы индивидуально. Наш пример — учебный, поэтому для простоты будем считать, что каждый раз потребитель недоплачивает 20 % необходимой суммы (= 0,8 х сумма счета), а оплату выполняет через 10 дней после выставления счета (= 10 + дата счета). Таким образом, для расчета параметров оставшихся столбцов можно воспользоваться формулами, в которых в качестве аргументов используются значения уже заполненных столбцов. В результате заполнения таблица должна иметь вид, подобный рис. 56 (стрелками показаны влияющие ячейки).

Продемонстрируйте результат преподавателю и сохраните рабочую книгу в личной папке.

Анализ результатов

Заполненная таблица пока мало информативна, так как содержит только исходные данные. Первое, что необходимо сделать, — это подвести итоги за квартал. Для этого в ячейку B12 вводим слово «Итого» и в последней строке таблицы просуммируем расход электроэнергии, ожидаемую и фактическую оплаты. Для этого необходимо переместиться в ячейку C12 и нажать кнопку Σ (*автосуммирование*) из группы команд *Библиотека функций* вкладки *Формула* или вкладка *Главная*, группа команд *Редактирование*. Пульсирующей рамкой Excel сам определил требуемый диапазон суммирования. Остается нажать *Enter*. Подобным же образом подводятся итоги в столбцах по счетам и по оплате (оплаты в апреле отнесем пока к первому кварталу).

Добавим в таблицу еще один столбец для расчета разницы между суммой в счете и оплатой. Назовем этот столбец «Долг». Его значения определим по формуле разности. В результате (после обрамления добавленного столбца) получим следующий вариант таблицы (рис. 57).

После расчета общего размера долга наступает этап персональной работы с потребителями. Для этого необходимо перегруппировать данные таблицы. Для начала воспользуемся сортировкой. Выделите любую ячейку таблицы, а затем на вкладке *Главная* в поле *Редактирование* выберите *Сортировка и фильтр* и затем команду *Настраиваемая сортировка*. Затем в окне *Сортировка* установите необходимые параметры (рис. 58) и нажмите кнопку OK. Здесь для именования параметров сортировки Excel использует названия из «шапки» таблицы.

	A	B	C	D	E	F	G	H
1	Оплата счетов за электроэнергию							
2	№	Потребитель	Месячный расход	Сумма в счете	Дата счета	Сумма оплаты	Дата оплаты	Долг
3	1	Иванов	10	4	31 янв	3,2	10 фев	0,8
4	2	Петров	20	8	31 янв	6,4	10 фев	1,6
5	3	Сидоров	30	12	31 янв	9,6	10 фев	2,4
6	4	Иванов	40	16	28 фев	12,8	10 мар	3,2
7	5	Петров	50	20	28 фев	16	10 мар	4
8	6	Сидоров	60	24	28 фев	19,2	10 мар	4,8
9	7	Иванов	70	28	31 мар	22,4	10 апр	5,6
10	8	Петров	80	32	31 мар	25,6	10 апр	6,4
11	9	Сидоров	90	36	31 мар	28,8	10 апр	7,2
12		**Итого**	450	180		144		36

Рис. 57. Вид таблицы на этапе расчета столбца «Долг»

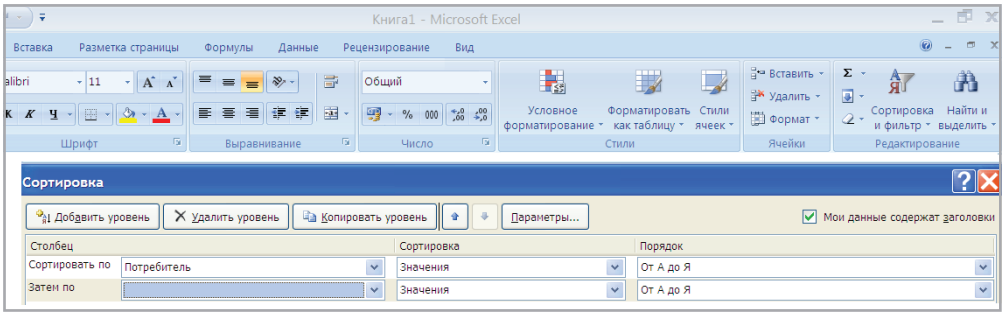


Рис. 58. Диалог сортировки

Следующий шаг — подведение итогов отдельно по каждому потребителю. Прежде всего необходимо удалить строку *Итого*, так как Excel подобную строку добавит сам. Сделать это можно, например, выделив всю строку с помощью щелчка левой кнопкой мыши на номере строки на «координатной оси», а затем, вызвав контекстное меню уже правой кнопкой мыши, выполнить в нем команду *Удалить*. После этого, оставаясь курсором на какой-либо ячейке таблицы, на вкладке *Данные*, в поле *Структура* выполнить команду *Промежуточные итоги* (рис. 59).

В открывшемся при этом диалоговом окне, прежде всего, устанавливаем необходимые параметры (рис. 59). Подведение итогов начинается после нажатия кнопки OK. Внимательно ознакомьтесь с результатами подведения промежуточных итогов. Если что-то получилось не так — есть возможность отменить последнюю команду. Для этого

на панели быстрого доступа (рядом с кнопкой *Office*) имеется кнопка Отменить. В результате таблица должна выглядеть примерно так, как показано на рис. 60.

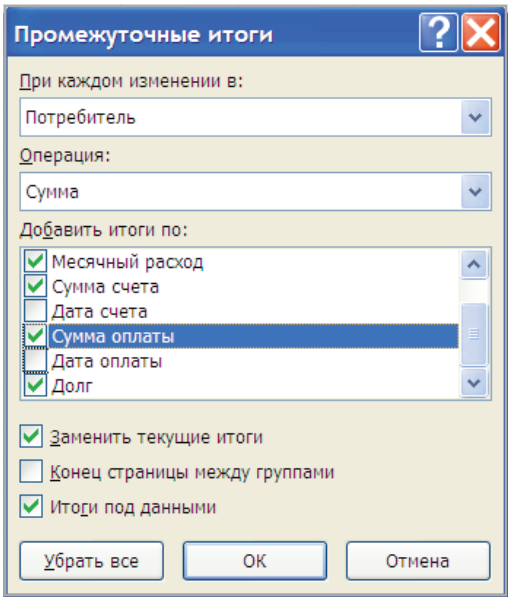


Рис. 59. Диалог *Промежуточные итоги*

	A	B	C	D	E	F	G	H
1	Оплата счетов за электроэнергию							
2	№	Потребитель	Месячный расход	Сумма счета	Дата счета	Сумма оплаты	Дата оплаты	Долг
3	1	Иванов	10	4	31 янв	3,2	10 фев	0,8
4	4	Иванов	40	16	28 фев	12,8	10 мар	3,2
5	7	Иванов	70	28	31 мар	22,4	10 апр	5,6
6		**Иванов Итог**	120	48		38,4		9,6
7	2	Петров	20	8	31 янв	6,4	10 фев	1,6
8	5	Петров	50	20	28 фев	16	10 мар	4
9	8	Петров	80	32	31 мар	25,6	10 апр	6,4
10		**Петров Итог**	150	60		48		12
11	3	Сидоров	30	12	31 янв	9,6	10 фев	2,4
12	6	Сидоров	60	24	28 фев	19,2	10 мар	4,8
13	9	Сидоров	90	36	31 мар	28,8	10 апр	7,2
14		**Сидоров Итог**	180	72		57,6		14,4
15		**Общий итог**	450	180		144		36
16								

Рис. 60. Вид таблицы после структуризации

Как следует из анализа полученной таблицы, Excel выполнил ее структурирование — организовал в виде древовидной системы. Корневым уровнем определена строка «Общий итог» (уровень номер 1). Уровнем ниже (номер 2) — строки с промежуточными итогами по каждому потребителю. И наконец, остальные строки составляют низовой дерева — уровень 3.

Воспользуемся полученной структурой, чтобы оставить в таблице только итоговые характеристики. Для этого щелкнем поочередно на кнопках управления детализацией уровня («–»). Результат должен быть как на рис. 61.

1 2 3		A	B	C	D	E	F	G	H
	1	Оплата счетов за электроэнергию							
	2	№	Потребитель	Месячный расход	Сумма счета	Дата счета	Сумма оплаты	Дата оплаты	Долг
+	6		**Иванов Итог**	120	48		38,4		9,6
+	10		**Петров Итог**	150	60		48		12
+	14		**Сидоров Итог**	180	72		57,6		14,4
–	15		**Общий итог**	450	180		144		36

Рис. 61. Итоговый вариант таблицы

Продемонстрируйте результат преподавателю, сохраните результат в личной папке.

Представление результатов

Итоговый вариант таблицы (рис. 61) вполне может быть использован в качестве отчетного. Но лучшую наглядность результатам придают диаграммы. Построим внедренную на лист диаграмму, отражающую баланс общей задолженности перед предприятием отдельных потребителей.

Работа по созданию диаграммы выполняется с помощью так называемого «мастера диаграмм» — специальной программы, в диалоге с которой задаются необходимые характеристики будущей диаграммы. Запускается мастер либо кнопкой на стандартной панели, либо кнопкой из группы *Диаграмма* на вкладке *Вставка*.

На первом шаге выбирается тип диаграммы — например, круговая и ее представление (простая, объемная). Далее Excel открывает закладку *Конструктор*, где можно выбрать нужную цветовую гамму, стиль диаграммы (группа кнопок *Стили диаграмм*), изменить тип диаграммы (группа *Тип*), выбрать различное оформление и представление данных на диаграмме (группа *Макеты диаграмм*). Для задания

области построения (т. е. на основании каких данных будет строиться диаграмма) нажмем кнопку *Выбрать данные* в группе *Данные* (рис. 62).

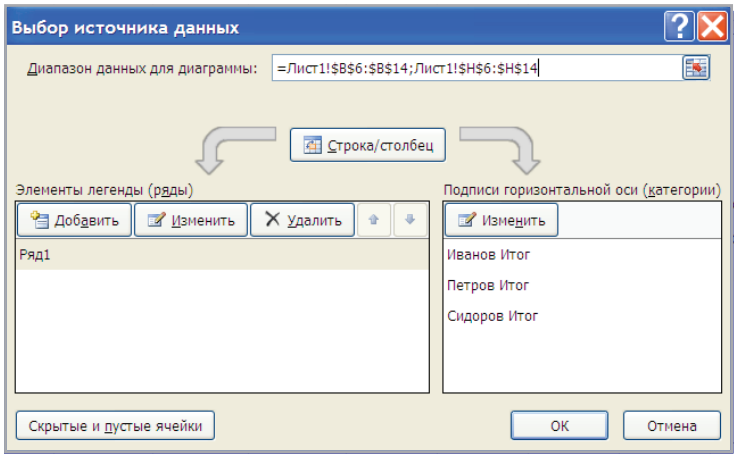


Рис. 62. Окно *Выбор источника данных*

Здесь задаются диапазоны значений для построения — в данном случае их два: для названий B6:B14 и для значений H6:H14. Выделить диапазоны проще мышкой на самой таблице, если предварительно немного отодвинуть окно мастера (зацепив его за заголовок).

Диапазоны являются несвязными. Первый выделяется обычным образом, а все последующие — при нажатой и удерживаемой клавишей *Ctrl*. Нажав кнопку ОК, вы получите круговую диаграмму. Измените ее макет так, чтобы можно было задать название диаграммы — «Квартальная задолженность» и отображалась доля (%) задолженности каждого потребителя (рис. 63). В случае необходимости можно вручную поправить расположение диаграммы на рабочем листе.

Рис. 63. Диаграмма

Самостоятельная часть

Выполните следующие пункты:

- добавьте единицы измерения для отпущенной электроэнергии и финансовых значений;
- добавьте в таблицы расчет % задолженности в оплате электроэнергии каждого потребителя;
- рассчитайте задолженности потребителей, не включая апрельские платежи в баланс по первому кварталу;
- измените вид диаграммы.

6. Работа с массивами и матрицами в Excel

Строго говоря, деление на матрицы и массивы достаточно условно. В Excel существует единственный тип векторных объектов — массивы, различающиеся размерностью. Наиболее естественный способ задания массивов — таблица чисел. Для обработки массивов в Excel могут быть использованы следующие функции:

МОБР (массив) — вычисление обратной матрицы;

МУМНОЖ (массив 1, массив 2) — умножение матрицы на матрицу или массив;

ТРАНСП (массив) — транспонирование матрицы;

МОПРЕД (массив) — вычисление определителя матрицы.

Несложно видеть, что результатом работы всех перечисленных функций, кроме последней, также является массив.

Элементарные матричные преобразования в Excel

Запустите табличный процессор Excel и удалите все листы рабочей книги, кроме первого. Выполняется это следующим образом. С помощью мыши выделите закладку «Лист2», затем при удерживаемой клавише *Shift* щелкните мышью по кнопке пролистывания книги до конца (книга чаше всего открывается с несколькими листами), а затем выполните команду *Удалить/Удалить лист* из группы команд *Ячейки* на вкладке *Главная*. Создайте числовую матрицу размерностью три строки на три столбца (рис. 64). Серым фоном в данной и последующих таблицах даны для ориентировки заголовки рабочего листа *Excel*. Имеет смысл сохранять заданное размещение объектов.

	A	B	C
1	Матрица А		
2	10	9	8
3	-1	-3	-6
4	-4	2	8

Рис. 64. Матрица А

Расчет определителя матрицы А

Выделите ячейку А5 и введите фразу «Определитель матрицы». Затем выделите ячейку А6 и на вкладке *Формулы*, группа команд *Библиотека функции*, выполните команду *Вставка Функции* либо на вкладке *Главная*, поле команд *Редактирование* откройте выпадающий список возле знака суммирования и выберите *Другие функции* (рис. 65).

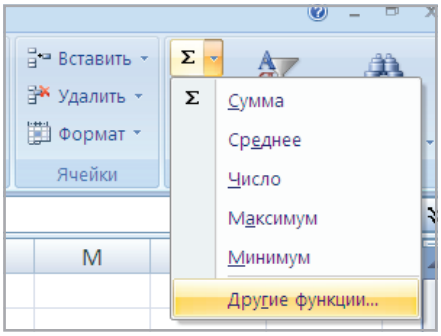


Рис. 65. Вставка функции

В диалоговом окне *Мастер функций* — шаг 1 из 2 выберите функцию МОПРЕД и нажмите кнопку OK (рис. 66). В окне *Мастер функций* — шаг 2 из 2 задайте диапазон ячеек, в котором размещена матрица А. Для этого мышью выделите ячейку А2 и, нажав левую клавишу мыши, растяните пунктирную рамку до ячейки С4. При этом в поле массив должна появиться запись «А2:С4». Если это действительно так, можно нажать кнопку OK. В ячейке А6 появится число 56.

Аналогичные действия вы можете выполнять с клавиатуры. Для этого в той ячейке, в которую необходимо вставить функцию, введите знак «=» и напишите название функции. Данный способ при правильном использовании может в значительной мере ускорить вашу работу как в Excel, так и в других приложениях.

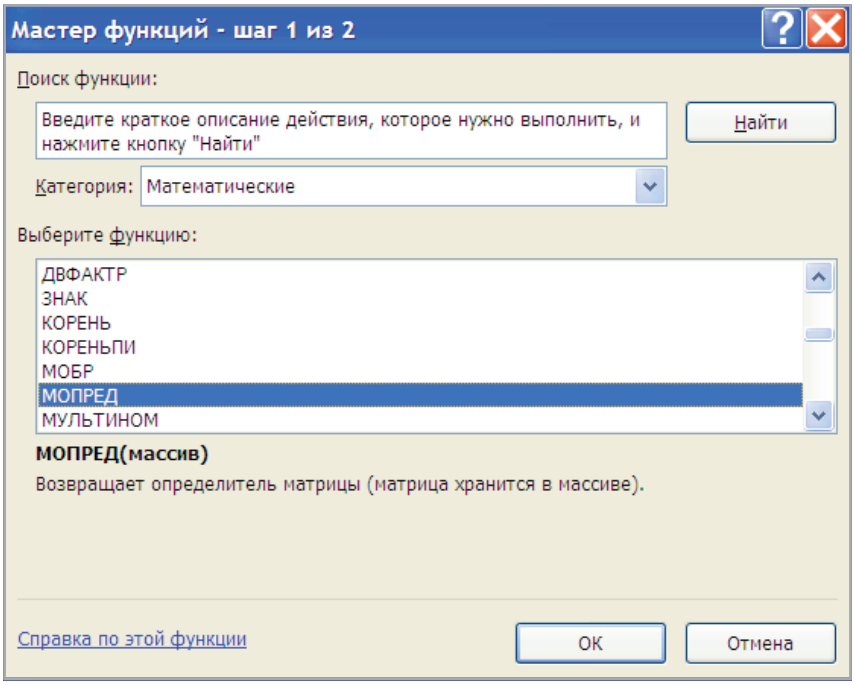


Рис. 66. Вызов мастера функций

Вычисление обратной матрицы

Поскольку определитель матрицы А ненулевой, она имеет обратную матрицу размерностью также 3 на 3. В первую очередь для обратной матрицы подготовьте место. Выделите ячейку А7 и наберите в ней фразу «Обратная матрица». Предполагая, что сама матрица разместится в ячейках А8:С10, выделите этот диапазон. Не снимая данного выделения, командой *Вставка Функции* активизируйте окно *Мастер функций шаг — 1 из 2.* Здесь следует выбрать функцию МОБР. Входной массив для нее задается также диапазоном А2:С4. Нажав кнопку ОК, вы запускаете вычисление обратной матрицы, но в качестве результата получаете только одно значение — в ячейке А8. Для того чтобы заполнились значениями и остальные ячейки обратной матрицы, необходимо установить курсор в строке формул (см. рис. 65) и нажать комбинацию клавиш *Ctrl+Shift+Enter*. Тем самым мы придаем массиву А8:С10 статус «формулы массива», т. е. массива, все элементы которого вычисляются по указанной формуле. Результат вашей работы должен быть близок представленному на рис. 67.

	A	B	C
1	Матрица А		
2	10	9	8
3	-1	-3	-6
4	-4	2	8
5	Определитель матрицы		
6	56		
7	Обратная матрица		
8	-0,21429	-1	-0,53571
9	0,571429	2	0,928571
10	-0,25	-1	-0,375

Рис. 67. Определитель и обратная матрица

Получение единичной матрицы

Для проверки правильности вычисления обратной матрицы выполните ее умножение на исходную — матрицу А. Для этого в ячейку Е1 впишите «Проверочная матрица» и выделите диапазон для результата произведения E2:G4. Затем с помощью мастера формул задайте в заданном диапазоне формулу МУМНОЖ. При этом в поле массив 1 задается диапазон A2:C4, а в поле массив 2 — диапазон A8:C10. Не забудьте после нажатия клавиши ОК придать результату статус «формула массива». Значения проверочной матрицы на первый взгляд подозрительны. Установите формат ячеек диапазона E2:G4 числовой формат без десятичных знаков. В итоге вы должны получить действительно единичную матрицу.

Транспонирование матрицы

Выполним транспонирование матрицы А. Для этого в ячейку Е7 запишите фразу «Транспонированная матрица», а в диапазоне E8:G10 разместите результат. Рабочий лист в итоге должен оказаться примерно таким, как на рис. 68.

Примечание: функция транспонирования называется ТРАНСП и находится в категории *Ссылки и массивы*.

Продемонстрируйте результат преподавателю, сохраните книгу в личной папке.

	A	B	C	D	E	F	G
1	Матрица А				Проверочная матрица		
2	10	9	8		1	0	0
3	-1	-3	-6		0	1	0
4	-4	2	8		0	0	1
5	Определитель матрицы						
6	56						
7	Обратная матрица				Транспонированная матрица		
8	-0,21429	-1	-0,53571		10	-1	-4
9	0,571429	2	0,928571		9	-3	2
10	-0,25	-1	-0,375		8	-6	8

Рис. 68. Фрагмент рабочего листа

Решение систем линейных уравнений

Наиболее востребованным применением Excel для работы с матрицами является решение систем линейных уравнений. Известно, что если A — матрица размерностью n строк на n столбцов, B — массив, состоящий из n элементов, а X — массив из n неизвестных, то система записывается как $A \cdot X = B$.

Умножением правой и левой частей данного матричного уравнения на матрицу A^{-1} (т. е. на обратную) находится решение $X = A^{-1} \cdot B$.

Средствами Excel решим следующую систему уравнений:

$$\begin{cases} 10x_1 + 9x_2 + 8x_3 = 2; \\ -x_1 - 3x_2 - 6x_3 = 1; \\ -4x_1 + 2x_2 + 8x_3 = 2. \end{cases}$$

Для начала удалите с вашего рабочего листа проверочную и транспонированную матрицы. В ячейках E2:E4 задайте массив правых частей, из чисел 2, 1, 2. Выделите диапазон E8:E10 для размещения в нем решения системы и найдите его с помощью умножения обратной матрицы на массив правых частей. Результат на вашем листе должен быть таким, как на рис. 69.

Как вы можете увидеть, матрица A является матрицей коэффициентов при искомых x, а массив правых частей, в свою очередь, является правой частью уравнений.

	A	B	C	D	E	F	G
1	Матрица *A*				Массив правых частей *B*		
2	10	9	8		2		
3	-1	-3	-6		1		
4	-4	2	8		2		
5	Определитель матрицы						
6	56						
7	Обратная матрица				Решение системы *X*		
8	-0,214	-1	-0,5357		-2,5		
9	0,5714	2	0,9286		5		
10	-0,25	-1	-0,375		-2,25		

Рис. 69. Решение системы линейных уравнений

Самостоятельная часть

По указанию преподавателя решите одну из систем линейных уравнений, выполните проверку каждого из уравнений по отдельности:

1. $\begin{cases} 7x_1 + 6x_2 + 4x_3 - 4x_4 = 4 \\ 3x_1 - 12x_2 + 5x_3 - 4x_4 = -3 \\ 11x_1 - 9x_2 + 10x_3 + 2x_4 = -4 \\ 2x_1 + 3x_2 - 6x_3 - x_4 = 15 \end{cases}$

2. $\begin{cases} x_1 + 2x_2 + 3x_3 - 4x_4 = 10 \\ 2x_1 - x_2 + 7x_3 - 8x_4 = -6 \\ 5x_1 - 9x_2 + 8x_3 + 2x_4 = -7 \\ 2x_1 + 3x_2 - 6x_3 - x_4 = 18 \end{cases}$

3. $\begin{cases} x_1 + 2x_2 + 3x_3 - 4x_4 = 10 \\ 2x_1 - 2x_2 + 9x_3 - 7x_4 = -6 \\ 2x_1 - 11x_2 + 3x_3 + 22x_4 = 0 \\ 4x_1 + 6x_2 - 2x_3 - 11x_4 = 100 \end{cases}$

4. $\begin{cases} x_1 + 2x_2 + 3x_3 + 4x_4 = -4 \\ 2x_1 + 5x_2 - 6x_3 + 4x_4 = 5 \\ 3x_1 - 6x_2 + 6x_3 - 14x_4 = 5 \\ 12x_1 - 14x_2 + 5x_3 = 8 \end{cases}$

5. $\begin{cases} x_1 + 2x_2 - 3x_3 + 4x_4 = 5 \\ 4x_1 + 2x_2 + 3x_3 + x_4 = 0 \\ 15x_1 - 16x_2 + 25x_3 + 34x_4 = 40 \\ 1x_1 + 58x_2 + 94x_3 = -19 \end{cases}$

6. $\begin{cases} x_1 + 2x_2 - 3x_3 + 4x_4 = 5 \\ 4x_1 + 2x_2 + 3x_3 + x_4 = 0 \\ 15x_1 - 16x_2 + 25x_3 + 34x_4 = 40 \\ 1x_1 + 58x_2 + 94x_3 = -19 \end{cases}$

7. $\begin{cases} 3x_1 + 6x_2 + 17x_3 + 22x_4 = 8 \\ x_1 + 14x_2 + 5x_3 + 17x_4 = 6 \\ 13x_1 + 9x_2 + 22x_3 + x_4 = 10 \\ x_1 + 10x_2 + 19x_3 + 92x_4 = 24 \end{cases}$

8. $\begin{cases} x_1 + 2x_2 + 3x_3 - 4x_4 = 10 \\ 3x_1 - 12x_2 + 5x_3 - 4x_4 = -3 \\ 11x_1 + 2x_2 - 5x_3 - 2x_4 = -6 \\ 2x_1 + 6x_2 - 4x_3 - 2x_4 = 16 \end{cases}$

9. $\begin{cases} 4x_1 - 2x_2 + 5x_3 - 3x_4 = 12 \\ 3x_1 + 2x_2 - 5x_3 + x_4 = -1 \\ 3x_1 - 13x_2 + 5x_3 + 12x_4 = -3 \\ 2x_1 + 3x_2 - 7x_3 - x_4 = 17 \end{cases}$

10. $\begin{cases} 5x_1 + 2x_2 - 4x_3 - 8x_4 = 8 \\ 3x_1 - 1x_2 + 9x_3 - 11x_4 = -9 \\ x_1 - 14x_2 + 10x_3 + 22x_4 = 4 \\ x_1 + 6x_2 - 3x_3 - 9x_4 = 6 \end{cases}$

11. $\begin{cases} 16x_1 + x_2 + 8x_3 + 5x_4 = 8 \\ x_1 + 14x_2 + 5x_3 + 17x_4 = 6 \\ 3x_1 + 12x_2 + x_3 + 5x_4 = 12 \\ 12x_1 + 6x_2 + 19x_3 + 60x_4 = 26 \end{cases}$

12. $\begin{cases} 2x_1 + x_2 + 17x_3 + 19x_4 = 10 \\ 16x_1 - 23x_2 + 19x_3 + 17x_4 = 4 \\ 13x_1 + 9x_2 + 22x_3 + x_4 = 10 \\ 31x_1 + 3x_2 + 19x_3 + 38x_4 = 24 \end{cases}$

13. $\begin{cases} 2x_1 + 6x_2 + 17x_3 + 5x_4 = 6 \\ x_1 + 12x_2 + 6x_3 + 11x_4 = 9 \\ 18x_1 + 6x_2 + 17x_3 + 4x_4 = 9 \\ 25x_1 + 5x_2 + 19x_3 + 50x_4 = 24 \end{cases}$

14. $\begin{cases} 6x_1 + 17x_2 + 15x_3 + 22x_4 = 6 \\ 16x_1 - 23x_2 + 19x_3 + 17x_4 = 4 \\ 13x_1 + 9x_2 + 22x_3 + x_4 = 10 \\ 5x_1 + 7x_2 + 19x_3 + 46x_4 = 20 \end{cases}$

7. Расчет сети постоянного тока в Excel

Рассматривается электрическая сеть, питающая потребителей постоянным электрическим током. Сеть (рис. 70) состоит из узлов (№ 1, 2, 3, 4) с заданными токами нагрузок и ветвей с активными сопротивлениями (Ом).

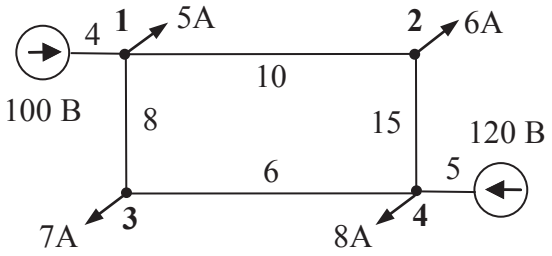


Рис. 70. Схема сети

Источниками электрической энергии (ЭЭ) являются устройства, поддерживающие заданное напряжение на своих выводах (источники ЭДС). Требуется определить уровни напряжений в нагрузочных узлах сети.

Расчет напряжений в узлах сети основан на законах Кирхгоффа и Ома. В матричном виде система уравнений выглядит следующим образом:

$$U = Y^{-1} \cdot (I - J),$$

где Y — матрица проводимостей ветвей, заданных в сименс (См); I — вектор нагрузочных токов узлов сети в амперах (А); J — вектор максимальных токов, которые могут обеспечить узлы с ЭДС в амперах; U — вектор искомых напряжений узлов, в вольтах (В).

Примечание. Проводимость ветви — это величина, обратная сопротивлению: [См] $= \left[{}^{1}\!/_{\text{Ом}} \right]$.

	Ток *I*, А
Узел 1	5
Узел 2	6
Узел 3	7
Узел 4	8

Рис. 71. Вектор токов нагрузки *I*

	ЭДС, В
Узел 1	100
Узел 2	–
Узел 3	–
Узел 4	120

Рис. 72. Вектор ЭДС

	Ток *J*, А
Узел 1	25
Узел 2	–
Узел 3	–
Узел 4	24

Рис. 73. Вектор токов генерации *J*

Для выполнения расчета напряжений узлов при помощи программы Excel необходимо записать исходные данные. Зададим вектор нагрузочных токов I (рис. 71).

Значения токов генерации определяются как произведение проводимости ветви, связывающей источник ЭДС, на величину ЭДС. В данной сети с источниками ЭДС связаны узлы 1 и 4. Зададим вектор ЭДС (Вольт) в соответствии с рис. 72.

Составляющие вектора *J* являются результатом умножения ЭДС на проводимость соответствующей ветви. Результат должен соответствовать таблице, показанной на рис. 73.

Следующий шаг — формирование матрицы проводимостей. Для начала задаем матрицу сопротивления ветвей электрической сети (без учета ветвей связи с источниками питания). Правило простое: если узел *i* связан с узлом *j*, то в ячейки матрицы на пересечении строки *i* и столбца *j*, а также строки *j* и столбца *i* записывается сопротивление ветви *ij* (рис. 74).

	Узел 1	Узел 2	Узел 3	Узел 4
Узел 1	–	10	8	–
Узел 2	10	–	–	15
Узел 3	8	–	–	6
Узел 4	–	15	6	–

Рис. 74. Матрица сопротивлений

Матрица проводимостей схемы является квадратной с размерностью, равной числу узлов. Формируется через расчетные выражения (в ячейки записываются формулы). Каждый элемент *ij* матрицы явля-

ется проводимостью ветви, соединяющей узел *i* с узлом *j*. Особым образом рассчитываются элементы матрицы проводимостей, расположенные на главной диагонали (*ii*, собственные проводимости).

В данном случае суммируются проводимости всех ветвей узла *i* (включая связь с источником ЭДС, если таковая имеется), а результат заносится в ячейку *ii* с обратным знаком (собственная проводимость узла всегда отрицательная). После задания соответствующих формул получается матрица проводимостей (рис. 75).

	Узел 1	Узел 2	Узел 3	Узел 4
Узел 1	-0,48	0,1	0,13	0
Узел 2	0,1	-0,17	0	0,07
Узел 3	0,13	0	-0,29	0,17
Узел 4	0	0,07	0,17	-0,43

Рис. 75. Матрица проводимостей

	Напряжение, В
Узел 1	52,35
Узел 2	15,23
Узел 3	26,75
Узел 4	49,55

Рис. 76. Вектор напряжений узлов

Вектор напряжений находится, как произведение матрицы, обратной матрице проводимостей на разность векторов $I - J$. После задания соответствующих формул напряжения узлов, представленные в числовом формате с точностью до двух цифр в дробной части, должны соответствовать рис. 76. Продемонстрируйте результат преподавателю, сохраните книгу в личной папке.

8. Регулирование режима сети в Excel

Анализ результатов предшествующей задачи позволяет сделать вывод, что узлы, удаленные от источников питания (в данном случае — это узлы 2 и 3), имеют заниженное значение напряжения относительно напряжений других узлов.

Для рассматриваемой сети постоянного тока одной из возможностей повысить уровни напряжений в нагрузочных узлах является увеличение напряжения в узлах — источниках ЭДС.

Для регулирования напряжения воспользуемся возможностями Excel, предоставляемыми сервисом *Подбор параметра*. Находится этот сервис на вкладке *Данные* в группе команд *Работа с данными* (рис. 77).

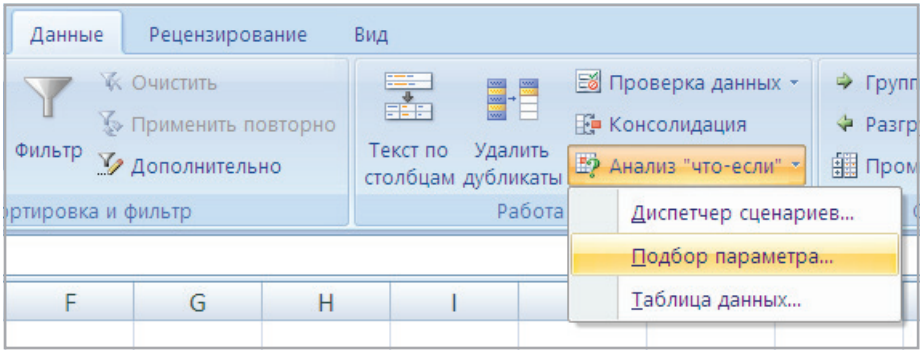


Рис. 77. Окно *Работа с данными*

На листе с результатами предыдущей работы добавьте еще один вектор, дублирующий значения рассчитанных напряжений узлов (операцией присвоения «=»). Дело в том, что в режиме автоматизированного подбора параметров ячейка должна содержать значение, а не формулу.

Для увеличения напряжения в узле 2 установите курсор на соответствующей ячейке дублирующего вектора напряжений и запустите ре-

жим *Подбор параметра*. Параметры подбора устанавливаются в диалоговом окне (рис. 78).

Здесь значение 20 В напряжения второго узла подбирается путем изменения напряжения источника ЭДС, связанного с узлом 1, то есть адрес E32 соответствует ячейке, где содержится значение ЭДС 1 (100 В).

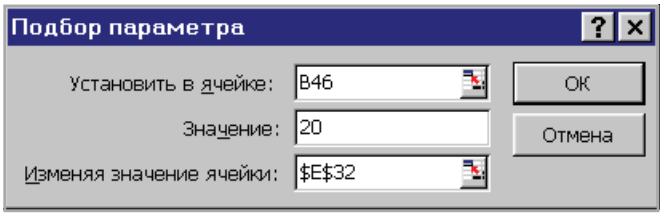


Рис. 78. Диалог *Подбор параметра*

Повторите подбор параметра (предварительно восстановив исходное состояние схемы) варьированием напряжения источника ЭДС, связанного с узлом 4. Выберите на ваш взгляд оптимальный вариант регулирования режима.

Задания для самостоятельного выполнения

- Выполните расчет электрического режима сети постоянного тока в соответствии с вариантом, предложенным преподавателем (рис. 79), табл. 14 и 15.

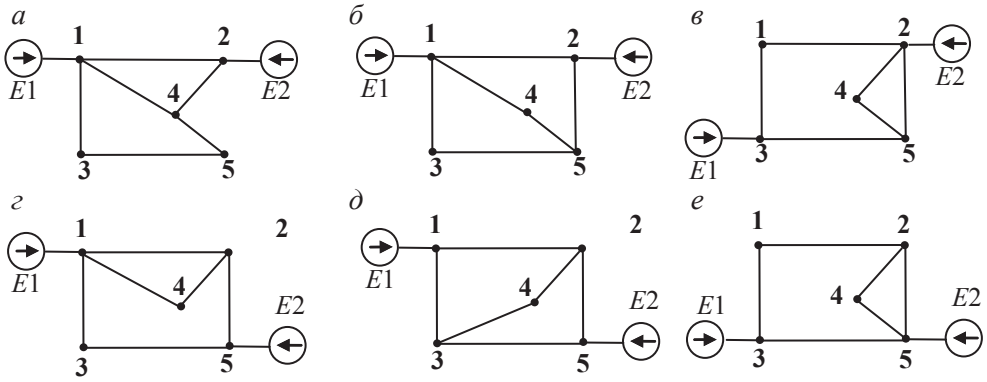


Рис. 79. Варианты схем

- Выполните оптимальное регулирование режима сети с целью повышения напряжения в наиболее удаленном от источников узле схемы на 20 %.
- Оформите отчет по данной работе в виде электронного документа, проиллюстрируйте решение данной задачи графическими средствами.

Таблица 14

№	Сопротивления ветвей, Ом							
	1–2	1–3	1–4	2–4	2–5	3–4	3–5	4–5
1	2	3	4	5	6	7	8	9
2	3	4	5	2	2	6	7	8
3	6	7	8	5	4	4	3	2
4	3	4	5	9	8	7	5	4
5	2	3	4	5	8	9	7	5
6	1	4	2	8	8	9	3	5
7	5	2	9	1	3	9	7	8
8	1	5	7	9	2	3	4	7
9	5	5	4	8	3	9	1	5
10	2	4	9	5	8	5	7	8
11	8	5	7	8	8	9	6	5
12	4	5	4	5	5	6	5	4
13	8	5	7	9	8	6	5	5

Таблица 15

№	Параметры источников генерации				Нагрузочные токи узлов, А				
	R_1, Ом	E_1, В	R_2, Ом	E_2, В	1	2	3	4	5
1	2	110	4	100	3	4	5	3	4
2	4	120	6	130	3	4	2	2	3
3	6	140	8	160	4	2	6	1	5
4	8	150	10	110	4	4	6	6	2
5	10	130	12	190	3	4	5	6	6
6	12	200	9	220	1	2	3	4	4
7	14	240	3	270	4	3	3	2	6
8	3	270	2	300	9	9	3	4	5
9	5	220	3	180	3	3	5	2	1
10	7	190	4	160	3	2	4	4	5
11	9	170	5	160	4	3	3	2	5
12	3	300	4	200	7	2	3	5	3
13	2	180	9	240	5	3	5	5	8

9. Решение транспортной задачи в Excel

Решение широкого круга задач энергетики основывается на оптимизации сложных зависимостей, формализованных в виде так называемой «целевой функции». Подобные функции можно записать для определения затрат на топливо для электростанций, потерь энергии при передаче ее потребителю и пр. В подобных случаях требуется найти минимум целевой функции, рассчитанный при определенных условиях. Если целевая функция линейно зависит от входящих в нее параметров, то такая задача сводится к задаче линейного программирования. Математические методы решения задачи линейного программирования изложены в соответствующей литературе. Целью настоящего занятия является использование специальных возможностей табличного процессора Excel при решении задачи линейного программирования, известной также под названием «транспортной задачи».

Постановка задачи

Предположим, что необходимо организовать перевозку угля от угольных бассейнов до электрических станций (рис. 80). Для упрощения задачи определим:

- одинаковый удельный расход топлива на всех станциях ($b_0 = 0{,}4$ т/МВт·час);
- одинаковую удельную стоимость перевозки тонны топлива на километр пути любого направления ($c_0 = 1$ у. е./тонну на км).

Решение задачи необходимо получить в виде распределения объемов перевозок топлива между угольными бассейнами и станциями при минимально возможных затратах на топливо и ограниченности годовых объемов добычи угля бассейнами.

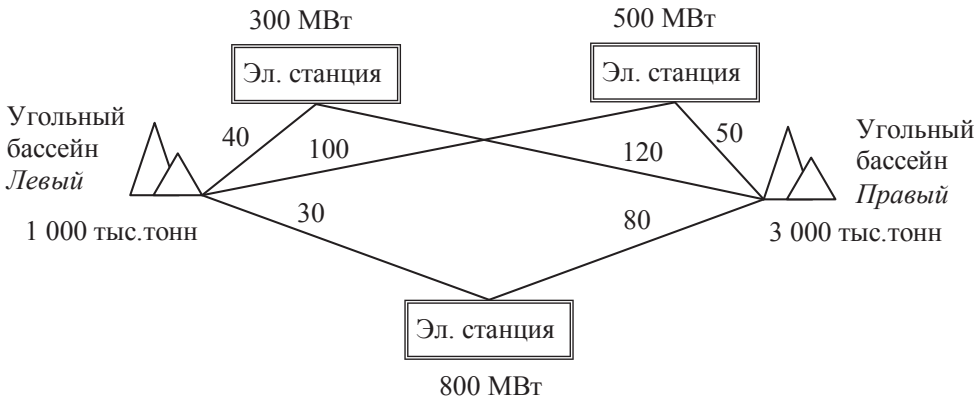


Рис. 80. Графическое представление постановки задачи

Подготовка рабочего листа

Запустите табличный процессор Excel и удалите все листы новой рабочей книги, кроме первого. В ячейке A1 поместите фразу «Транспортная задача», а в ячейке A2 — «Электрические станции».

Станционный блок

Информацию по электростанциям удобно сформировать в виде табл. 16.

Таблица 16

	A	B	C	D	E
1	Транспортная задача				
2	Электрические станции		Первая	Вторая	Третья
3	Руст., МВт		300	500	800
4	Т уст., часов		6800	5500	6500
5	Годовая выработка эл. эн, МВт час		2040000	2750000	5200000

В данной и последующих таблицах для ориентировки на сером фоне даны заголовки столбцов и строк рабочего листа Excel.

Каждая ячейка строки 5 таблицы должна содержать формулу произведения значений вышерасположенных ячеек, например, для ячейки C5 = C4*C3. Выполняется это следующим образом. Выделяется ячейка (в данном случае C5), нажимается клавиша «=» и клавишей ↑ клавиатуры пульсирующая рамка перемещается на ячейку — первый со-

множитель (в данном случае — С4) или та же процедура может быть выполнена при помощи мыши. При этом в строке формул появляются действительные координаты данной ячейки. После этого нажимается клавиша «*» (умножение) и то же самое повторяется для второго сомножителя — ячейки С3.

Если вид вашей таблицы несколько отличается от приведенного (см. табл. 12), воспользуйтесь командой вкладки *Главная*, группа *Ячейки*, команда *Формат — Размер ячейки — Автоподбор Ширины Столбца*, выделив предварительно всю таблицу.

Заголовок, помещенный в ячейку А1, можно разместить посередине таблицы, если выделить ячейки диапазона А1:Е1 и с помощью команды меню *Формат Ячейки* на закладке *Выравнивание* включить опцию *Центрировать* по выделению.

Транспортный блок

Расположение станций и бассейнов указано на рис. 80. Для дальнейшей работы нам понадобится табл. 17, которую сформируем по принципу «расстояния бассейны–станции». Начнем данную таблицу в строке 6.

Таблица 17

	A	B	C	D	E
6	Расстояния станции–бассейны (км)				
7	Бассейн Левый		40	100	30
8	Бассейн Правый		120	50	80

В таблице расстояний используется то же деление столбцов между станциями, что и в первой таблице.

Блок затрат на топливо

Последняя таблица рабочего листа выполняет объединяющую функцию. Заготовим таблицу (табл. 18) для нахождения затрат на топливо, поместив соответствующий заголовок в ячейке А9.

Таблица 18

	A	B	C	D	E
9	Затраты на топливо				
10	Потребность станции, тыс.т.				
11	Вывезено из бассейна Левый				
12	Вывезено из бассейна Правый				
13	Доставлено на станцию				
14	Затраты для станции				
15	Итоговые затраты на топливо				

Заполнение таблицы

В ячейки строки 10 должны быть вписаны формулы произведения годовой выработки электроэнергии станцией (ячейки строки 5) на удельный расход топлива (одинаковый для всех станций и равный 0,4 тонны на 1 МВт·час). Рекомендуется третьим сомножителем добавить 0,001 для того, чтобы потребность в топливе считать в тысячах тонн.

Ячейки строки 11 и 12 первоначально можно заполнить нулями. Здесь по окончании расчетов программа должна сама разместить оптимальные объемы перевозок топлива на каждую станцию из соответствующего бассейна. Рекомендуется выделить все ячейки данных строк и выполнить команду *Формат Ячейки* и на закладке *Число* установить *числовой* формат ячеек без десятичных знаков.

В ячейках строки 13 помещаем формулы суммирования значений ячеек строк 11 и 12 соответствующего столбца. Ячейки строки 14 содержат для каждой станции формулы произведений объема перевозки из соответствующего бассейна, расстояния до соответствующего бассейна и стоимости перевозки. Например, для станции «первая» (ячейка C14) формула должна иметь вид «=C7*C11+C8*C12», поскольку удельные затраты на перевозку положены равными 1 у. е. Подобным же образом подготавливаются формулы для ячеек D14 и E14.

В строке 15 нам понадобится только одна ячейка — B15. В нее следует поместить формулу суммирования содержимого ячеек C14, D14 и E14. Именно ячейка B15 и является целевой для решения задачи оптимизации перевозок топлива.

Блок добычи топлива

Небольшую таблицу (табл. 21), характеризующую возможности угольных бассейнов по годовой добыче топлива (тысяч тонн), разместим в строках 16–18 рабочего листа.

Таблица 21

	A	B
16	Добыча топлива	
17	Левый	1000
18	Правый	3000

Процесс оптимизации

Выделите ячейку B15 и на вкладке *Данные* в группе *Анализ* выполните команду *Поиск решения* (рис. 81). В поле *Установить целевую ячейку* диалогового окна вы должны увидеть запись — B15. Знак доллара в данном случае означает, что координаты ячейки не могут измениться при каких-либо изменениях рабочего листа. Переключатель *Равной* следует установить в позиции *Минимальному значению*. В поле *Изменяя ячейки* задайте диапазон C11:E12 (выделяя ячейки).

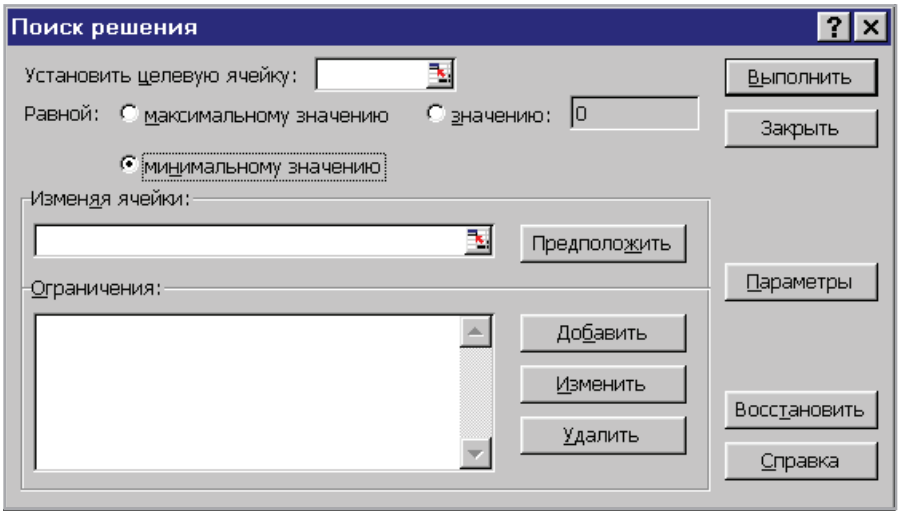


Рис. 81. Диалог *Поиск решения*

Если команды (группы команд) нет на ленте вкладки *Данные*, необходимо через кнопку *Office* изменить параметры *Excel*, *Надстройки*

и перейти к управлению надстройками Excel, выбрав в предлагаемом диалоговом окне *Поиск решения*.

Последним шагом поиска решения рассматриваемой задачи является задание ограничений.

Первым ограничением является положительность объемов перевозок топлива на станцию. В данном случае вы должны нажать на кнопку *Добавить* и в окне *Добавление ограничений* в левом поле задать диапазон ячеек C11:E12, выбрать ограничение >= и в правом поле поместить 0.

Следующими ограничениями учитывается годовая добыча топлива в бассейнах. Предварительно добавим в ячейку F11 формулу суммирования ячеек C11, D11, E11 — годовая отгрузка топлива с бассейна *Левый*. То же проделаем с ячейкой F12 — для бассейна *Правый*. После этого вводим ограничения F11<=B17 и F12<=B18 — действительно, с бассейна невозможно вывезти больше топлива, чем на нем добывается.

Последним ограничением необходимо задать условие удовлетворения запросов станций в топливе: C10: E10<=C13: E13. Символ двоеточия в формулах означает диапазон, таким образом данная формула эквивалентна трем неравенствам: C10<=C13, D10<=D13 и E10 <= E13. Последним шагом решения задачи является собственно запуск процесса оптимизации нажатием кнопки *Выполнить*. В специальном окне *Результаты поиска решения* программа сообщает о завершении процесса. Нажав кнопку OK, вы можете перенести результаты на рабочий лист.

В качестве самостоятельного упражнения решите задачу линейного программирования. Варианты заданий приведены в Приложении 1 (выбор по указанию преподавателя). Оформите отчет в виде электронного документа.

10. Основы работы в Mathcad

В последнее время для проведения научно-технических расчетов уже в меньшей степени требуется применение традиционных языков программирования или электронных таблиц и все чаще в инженерных расчетах встречается использование специальных математических программ типа Mathematica, MatLab, Maple, Mathcad, Gauss и других. Такие математические пакеты позволяют специалистам очень быстро найти решение рассматриваемой проблемы, не углубляясь в тонкости и особенности языков программирования [4, 5].

Основные достоинства работы в среде такой специализированной программы (Mathcad, Mathematica, MatLab) заключаются в следующем:

- данные пакеты являются Windows-приложениями, поэтому существует возможность передавать данные в среду другой программы (через буфер обмена или OLE-технологию) и решать поставленную задачу по частям;
- математические выражения могут быть записаны в общепринятой редакции. Например, для нахождения суммы членов ряда записывается известный знак суммирования, а не создается циклическая подпрограмма;
- в основе указанных пакетов лежит мощный математический аппарат, позволяющий решать задачи решения алгебраических уравнений и систем как линейных, так и нелинейных. Решение обыкновенных дифференциальных уравнений и систем, статистическая обработка данных, работа с векторами и матрицами, оптимизационные задачи также не вызывают больших трудностей при их решении.

Пустой Mathcad-документ — это «лист бумаги», на котором пользователь с помощью клавиатуры и мыши может писать математические выражения в том виде, как это ему уже привычно. С точки зрения

пользователя, документ — это некоторая область, на которой можно размещать блоки трех основных типов: *математические выражения*, *текстовые фрагменты* и *графические обл*асти.

На рис. 82 приведены некоторые (далеко не все) возможности Mathcad, предлагаемые на панели математических инструментов. Поскольку пользователю редко требуются сразу все, ненужную палитру математических инструментов в любой момент времени можно свернуть, нажав на «крестик» в правом верхнем углу палитры. (В зависимости от версии программы вид рабочего стола может иметь незначительные отличия).

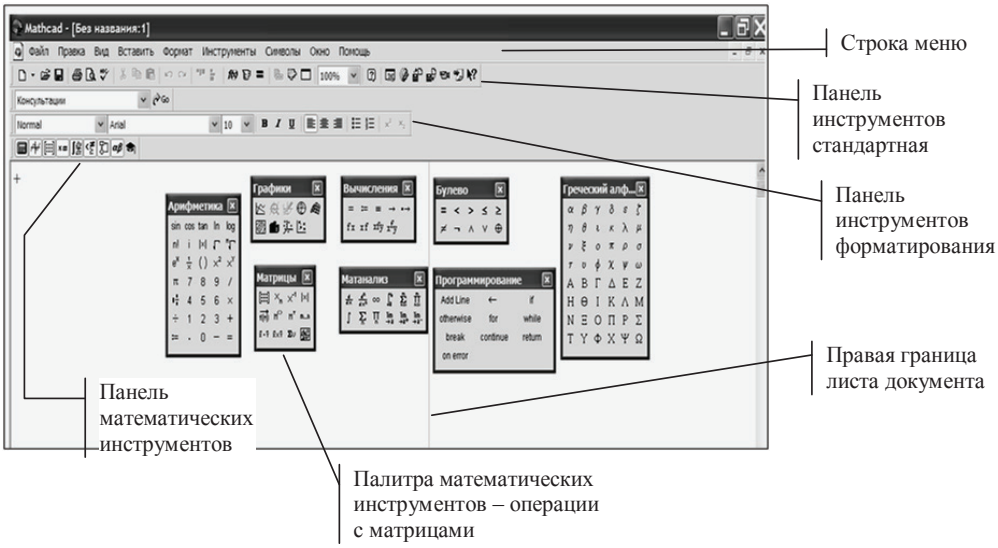


Рис. 82. Рабочий стол Mathcad

Прежде чем начинать работу, необходимо знать несколько правил или особенностей Mathcad:

1. В качестве разделителя целой и дробной части числа используется точка.

2. Mathcad прочитывает весь документ дважды слева направо и сверху вниз. При первом проходе выполняются все действия, предписанные глобальным оператором присваивания, а при втором — производятся действия, предписанные локальным оператором присваивания и отображаются все необходимые результаты вычислений.

По умолчанию, нажимая на кнопки клавиатуры и клавиши мыши, пользователь задает математические выражения. К основным элементам математических выражений Mathcad относятся типы данных, операторы, функции и управляющие структуры.

Ввод и редактирование

- Подготовка документа. Найдите на рабочем столе ярлык и с его помощью зайдите в среду Mathcad. Сохраните предлагаемый документ как «work1.xmcd» на рабочем диске *D*:\в личной папке с номером группы (расширение «*.xmcd» будет установлено автоматически).
- Для того чтобы освоить применение некоторых кнопок и клавиатуры для ввода и редактирования математических выражений, выполните упражнения, приведенные в табл. 20.

Таблица 20

Установив щелчком мыши визир (красный крестик +) на свободном месте документа и напечатав на клавиатуре 2+1/2=	Получите $2+\frac{1}{2}=2.5$
Установив визир (+) щелчком мыши на свободном месте и напечатав: (9+3)/(2*4–5)=	Получите $\frac{(9+3)}{(2\cdot 4-5)}=4$
Напечатав это (предварительно установив визир): 3+\2=	Получите $3+\sqrt{2}=4.414$
Установив визир и напечатав: 0.63log(x^2)-log(1000)=	Получите $0.63\log(x^2)-\log(1000)$

В версиях программы, начиная с Mathcad 2001, доступно представление результата в дробном формате. Щелкните на правой части первого равенства, предлагаемого для упражнений. Затем сделайте двойной щелчок левой клавишей мыши на результате вычисления и выберите на вкладке *Результат* (ResultFormat) → *Формат* (NumberFormat) опцию *Доля* (Fraction) и нажмите OK. Получите ответ в виде неправильной дроби. Доступ к этой вкладке можно получить и через основное (верхнее) меню, *Формат* (Format) → *Результат* (Result).

При работе пользователю удобнее оперировать не числовыми значениями, а некоторыми переменными величинами. В Mathcad со-

держится небольшая группа особых объектов, которые нельзя отнести ни к классу констант, ни к классу переменных, значения которых определены сразу после запуска программы. Их правильнее считать *системными переменными*, имеющими предопределенные системой начальные значения. Изменения значений системных переменных производят во вкладке *Инструменты* (Tools), *Опции таблицы* (Worksheet Options), вкладка *Built–in Variables*. Обычные переменные, задаваемые пользователем, отличаются от системных (встроенных) тем, что они должны быть предварительно определены, т. е. им необходимо хотя бы однажды присвоить какое-то значение.

Повторите «пример 1», приведенный на рис. 83. Здесь задаются две переменные a и b (имена переменных задаются произвольно по желанию пользователя). В качестве оператора присваивания используется знак [:=] (в табл. 21 приведены «горячие клавиши», позволяющие быстро задать тот или иной оператор), тогда как знак [=] отведен для вывода значения константы или переменной. Если переменной присваивается начальное значение с помощью оператора [:=], такое присваивание называется локальным. До этого присваивания переменная не определена и ее нельзя использовать. С помощью знака [≡] можно обеспечить глобальное присваивание (рис. 83). Комментарии задаются в виде «текстовых фрагментов», представляют собой куски текста, которые пользователь хотел бы видеть в своем документе. Текстовые комментарии лучше задавать как отдельную область документа, при помощи вкладки *Вставка* (Insert) → *Область текста* (TextRegion) или комбинации клавиш [Shift]" (двойная кавычка).

Таблица 21

Оператор	Пример использования в Mathcad	«Горячая клавиша»	Описание
Присвоить значение переменной, элементу массива, определить функцию пользователя	□:=□ a:=b+c $M_{i,j}$:=5.2	:	Присваивает значение переменной, определяет функцию, видимую правее и ниже данного оператора
Присвоить значение глобальной переменной	a ≡ b+c	~	Присваивает значение переменной, определяет функцию, видимую во всем документе

Окончание табл. 21

Оператор	Пример использования в Mathcad	«Горячая клавиша»	Описание
Вычисление числового значения	□=□ a=23.5 f(x)=5	=	Вычисляет и выводит на экран числовое значение переменной (если переменная была не определена, преобразуется в оператор присваивания)
Нижний индекс	B_k	[	Возвращение индексированного элемента массива
Степень	z^w	^	Возводит параметр *z* в степень *w*
Деление	$\frac{X}{y}$	/	Деление выражения X на скаляр *y*, не равный 0
Задание диапазона	1,3..n x,x+2..N	;	Задание ряда значений от начального значения до крайнего, с некоторым шагом

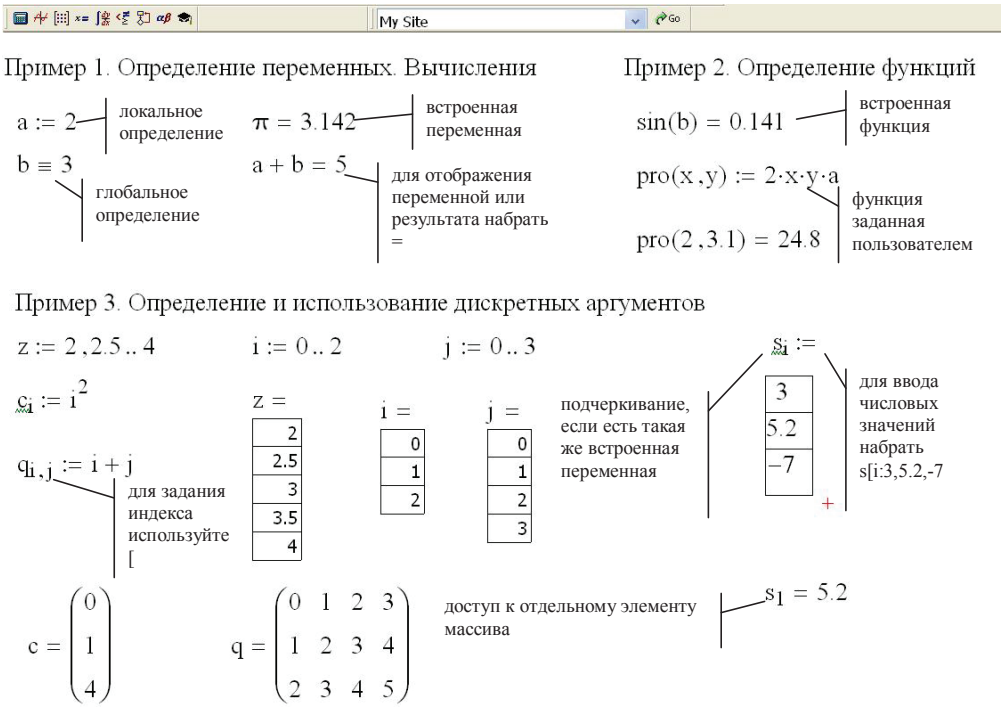


Рис. 83. Пример задания выражений в Mathcad

Этап 1

Выполнить вычисления. Это и все остальные задания снабдить комментариями и отразить в отчете.

$\sqrt{100}$	$\lvert -10 \rvert$	$10!$	$2 \cdot 53^2 / \left(160 - \sqrt{44}\right)$
$\pi/6$	$e^{-1/2}$	$\sin(\pi/3)$	$\cos(0.454)$

Для того чтобы на экране в документе Mathcad были видны области математических выражений и области текстовых комментариев на сером фоне, необходимо в пункте меню выбрать *Вид* (View) → *Границы* (Regions).

Этап 2

- Определить переменные и выражения:

  $a := 3.4$ $\quad b := 6.22$ $\quad c \equiv 0.149$ (глобально)

  $$z := \frac{2 \cdot a \cdot b + \sqrt[3]{c}}{(a^2 + b^{a+c}) \cdot c} \qquad d := a \cdot b + 2 \cdot c \cdot \frac{\sqrt[3]{4 \cdot b^2}}{|a - b| \cdot e^c} \qquad N := e^{\sin(c)} \cdot \cos\left(\frac{a}{b}\right)$$

- Вычислить выражения.
- С помощью команды *Формат* (Format) → *Результат* (Result) → *Формат номера* (Number Format) → *Число знаков* (Number of decimal places) изменить точность отображения результатов вычисления глобально.
- Повторите пример использования единиц измерения, приведенный на рис. 84.

Выполняя сложный расчет в именованных единицах, пользователь может воспользоваться встроенными в Mathcad единицами измерения или создать свои. Использование именованных единиц «не разрешает деление метров на яблоки» и в ряде случаев помогает пользователю.

Для присваивания величине размерности за ней ставится знак умножения (необязательно) и вводится название соответствующей размерности. Или можно нажать на панели часто используемых команд кнопку с изображением мерной кружки, тогда на дисплее появится окно со списками физических величин (длина, время, скорость и т. д.) с соответствующими им размерностями (метр, секунда, километр в час и т. д.), одна из которых вставляется в документ.

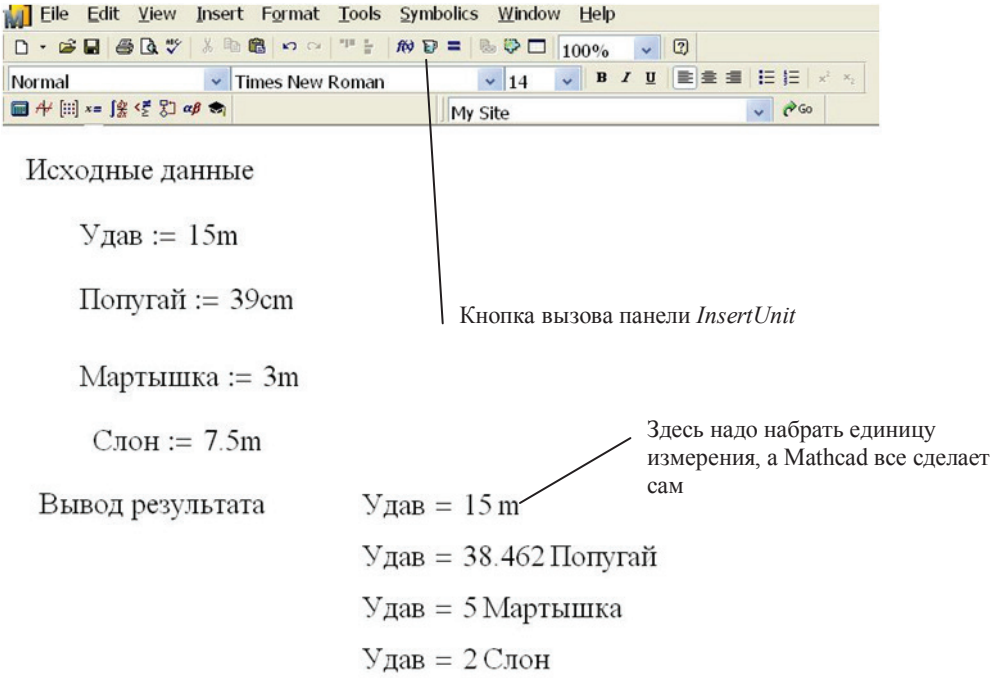


Рис. 84. Пример использования единиц измерения

Этап 3

- Вывести на экран значение системной константы π и установить максимальный формат ее отображения локально (найти ее можно на панели инструментов *CalculatorToolbar*). Опция меню *Format* позволяет управлять представлением результатов вычислений, количеством отображаемых знаков после десятичной точки. По умолчанию результат округляется до третьего знака после десятичной точки.
- Повторить «пример 2», показанный на рис. 83.

В данном примере производятся действия с функцией или выражением, согласно которому проводятся некоторые вычисления с аргументами и определяется его числовое значение. Следует особо отметить разницу между аргументами и параметрами функции. Переменные, указанные в скобках после имени функции, являются ее аргументами и заменяются при вычислении функции значениями из скобок. Переменные в правой части определения функции, не указанные скобках

в левой части, являются параметрами и должны задаваться до определения функции.

Главным признаком функции является «возврат значения», т. е. функция в ответ на обращение к ней по имени с указанием ее аргументов должна возвратить свое значение. Функции в пакете Mathcad могут быть «встроенные», т. е. заблаговременно введенные разработчиками и «определенные пользователем».

Способы вставки встроенной функции:

— выбрать пункт меню *Вставить* (Insert) → *Функция* (Function);

— нажать комбинацию клавиш [Ctrl] E или щелкнуть на кнопке  .

Этап 4

Выполнить следующие операции с комплексными числами:

- Задать комплексные числа:
  $Z := -3 + 4i, \quad Z_1 := 10 + 3i, \quad Z_2 := 8 + 9i.$
  Здесь «1», «2» являются подстрочными символами (по аналогии с $S_{расч}$, k_{max_1}). Не являются индексами! Задание такого символа возможно при помощи клавиши «точка».
- Определить результат следующих операций над комплексными числами:

$\lvert Z\rvert$	Re (Z)	Im (Z)	arg (Z) (в градусах)	$\sqrt{Z}$	$2\cdot Z$
$\sqrt{-5}$	$Z_1 + Z_2$	$Z_1 - Z_2$	$Z_1\cdot Z_2$	Z_1/Z_2	

Этап 5

- Повторить пример 3 с рис. 83.

Дискретные аргументы — особый класс переменных. Эти переменные имеют ряд фиксированных значений либо целочисленных (задаются 1-м способом), либо в виде чисел с определенным шагом, меняющихся от начального значения до конечного (2-й способ).

1) *«Имя переменной» := «начальное значение»..«конечное значение»*.

Здесь $\boxed{..}$ — символ, указывающий на изменение переменной в заданных пределах (вводится клавишей «;»). Если по модулю начальное значение меньше конечного, то шаг переменной будет равен «+1», иначе «–1». Например, $i := 1..10$ (i принимает значения 1, 2, 3...10).

2) *«Имя переменной» := «начальное значение», «начальное значение + шаг изменения»..«конечное значение»*.

Здесь необходимо знать шаг изменения переменной, он может быть как положительным, так и отрицательным. Например, запись $k := 20,15..0$ означает, что k принимает значения 20, 15, 10, 5, 0 (шаг изменения равен –5).

Дискретные аргументы значительно расширяют возможности Mathcad, позволяя выполнять многократные вычисления или циклы с повторяющимися вычислениями, формировать векторы и матрицы.

Массив — имеющая уникальное имя совокупность конечного числа числовых или символьных элементов, упорядоченных некоторым образом и имеющих определенные адреса. В пакете Mathcad используются массивы двух наиболее распространенных типов, а именно — одномерные (векторы) и двумерные (матрицы). Порядковый номер элемента, который является его адресом, называется *индексом*. Индексы могут иметь только целочисленные значения. Они могут начинаться с нуля или единицы в соответствии со значением системной переменной *ORIGIN* (по умолчанию *ORIGIN* = 0).

Векторы и матрицы можно задавать различными способами:

— с помощью команды *Вставить* (Insert) → *Матрица* (Matrix) или комбинации клавиш [Ctrl] M, или щелчком на кнопке  панели *Матрица*, заполнив массив пустых полей для не слишком больших массивов;

— с использованием дискретного аргумента, когда имеется некоторая явная зависимость для вычисления элементов через их индексы.

- Выполнить следующие операции:
  Задать $i := 1..10$ $x := 2$.
  Определить результат следующих выражений:

$$\sum_i i, \qquad \int_0^{0.4} x^2 \log(x+2)dx, \qquad \prod_i (i+1),$$

$$\int_{0.8}^{1.2} \frac{\cot(2x)}{(\sin(20\cdot x))^2}dx, \qquad \frac{d}{dx}x^5, \qquad \frac{d}{dx}\sin(x).$$

Этап 6

Графические области делятся на три основных типа — двумерные графики, трехмерные графики и импортированные графические образы. Двумерные и трехмерные графики строятся самим Mathcad на основании обработанных данных.

- Постройте график, как показано на рис. 85.

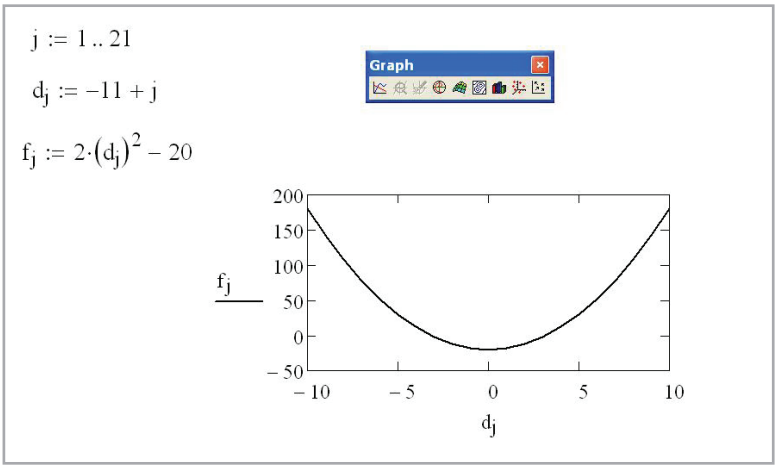


Рис. 85. Фрагмент окна Mathcad. Пример построения декартового графика

Для создания декартового графика необходимо:

— установить визир в пустом месте рабочего документа;

— выбрать команду *Вставить* (Insert) → *Графики* (Graph) → *X-Y зависимости* (X-Y Plot) или нажать комбинацию клавиш [Shift]@, или щелкнуть кнопку  панели *GrafToolbar*. Появится шаблон декартового графика. В средней метке под осью *X* задается одна или несколько (через запятую), но не более 10 независимых переменных. Слева от вертикальной оси *Y* в средней метке задаются зависимые переменные, функции или математические выражения, разделенные запятыми. Чтобы построить график, необходимо установить визир вне его области.

- Определить векторы *d*, *S* и *R* через дискретный аргумент *i*. Отобразить графически заданные переменные S_i и R_i как функции d_i. Здесь *i* изменяется от 0 до 4 с шагом «+1»; d_i принимает значения 0.5, 1, 1.5, 2, 2.5 — записать как зависимость значения d_i от вели-

чины i; S_i для каждого i равно 3.3, 5.9, 7, 6.3, 4.2 — задается отдельными значениями как в примере 3 на рис. 83; R_i соответственно 2, 3.9, 4.5, 3.7, 1.2 — задается аналогично S_i.

- Ответить на вопросы:
  — к чему приводит выбор закладки «логарифмическая ось» в окне форматирования графика?
  — как меняется внешний вид графика при изменении стиля оси?
  — что означает «автомасштабирование» осей?
- Оформить график в соответствии с указанным преподавателем вариантом.
  1. Показать легенду, переименовать графики в легенде как graf_1 для S_i и graf_2 для R_i. Установить для S_i: пунктирную линию, цвет — синий, толщина — 2, символ — ×. Для R_i: цвет — сиреневый, тип — «по шагам».
  2. Установить для S_i штрих-пунктирную линию, цвет — коричневый, толщина — 3, символ — o. Для R_i: цвет — голубой, тип — «гистограмма». Цвет линий сетки — темно-серый, количество линий сетки по обеим осям равно 10.
  3. Показать легенду, переименовать графики в легенде. Установить для S_i сплошную линию, цвет — красный, толщина — 2, символ — ◊. Для R_i: цвет — черный, тип — «точечная», символ — □. Показать линии сетки по оси Y, количество линий 5.
  4. Установить для S_i штриховую линию, цвет — зеленый, толщина — 3, символ — o. Для R_i: цвет — сиреневый, тип — «по основанию», символ — □. Цвет линий сетки — черный.

  Для оформления графика (изменения вида зависимости, видимости сетки и т. п.) необходимо:
  — щелкнуть левой клавишей мыши на графике, чтобы выделить его. Затем щелкнуть правой клавишей мыши, при этом появится контекстное меню, в котором необходимо выбрать команду *Формат* (Format) (появится диалоговое окно *Formatting Currently Selected X-Y Plot*, рис. 86).

  Вызов диалогового окна возможен и при двойном щелчке левой клавишей в области построения;
  — чтобы нанести линии сетки на график, выбрать *Оси X–Y* (X–Y Axes) → *линии сетки* (Grid Lines), размер сетки задается по осям;
  — отобразить легенду: *След* (Traces) → *Скрыть легенду* (Hide Legend);

— отформатировать график так, чтобы в каждой узловой точке графика функции стоял знак вида □, ×, o. Используется *След* (Traces) → *Символ* (Symbol), отображение в виде линии, точек или гистограммы задается через *След* (Traces) → *Тип* (Type).

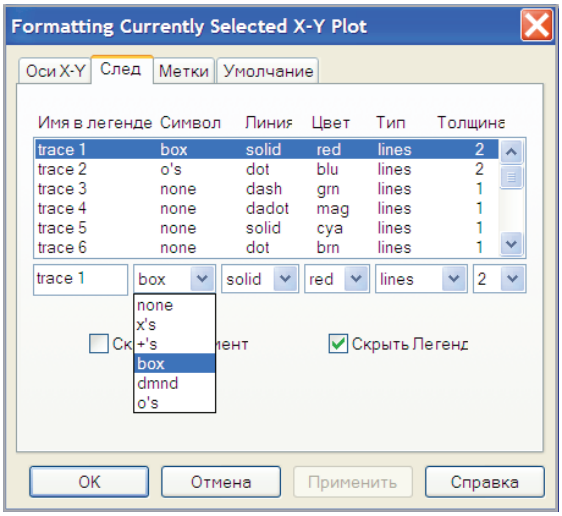


Рис. 86. Диалоговое окно форматирования графика

Этап 7

- Построить декартовы *X–Y Зависимость* и *Полярные* (Polar Plot) графики следующих функций:
  $X(\alpha) := \cos(\alpha) \cdot \sin(\alpha)$;
  $Y(\alpha) := 1.5\cos(\alpha^2) - 1$;
  $P(\alpha) := \cos(\alpha)$.
  Для этого необходимо определить α как дискретный аргумент на интервале от 0 до 2·π с шагом π/30.
- Определить по графику *X–Y Зависимость* координаты любой из точек пересечения графиков $Y(\alpha)$ и $P(\alpha)$. Для этого необходимо:
  — выделить график и выбрать из контекстного меню *Zoom* (появится диалоговое окно *X–Y Zoom*) для увеличения части графика в области точки пересечения;
  — на чертеже выделить пунктирным прямоугольником окрестность точки пересечения графиков $Y(\alpha)$ и $P(\alpha)$, которую нужно увеличить;

— нажать кнопку с изображением лупы, чтобы перерисовать график;
— чтобы сделать это изображение постоянным, выбрать OK;
— выбрать из контекстного меню *Трассировка* (Trace) (появится диалоговое окно *X–Y Trace*);
— внутри чертежа нажать кнопку мыши и переместить указатель мыши на точку, чьи координаты нужно увидеть;
— выбрать *Copy X* (или *Copy Y*), на свободном поле документа набрать Xp := (или Yp: =) и выбрать пункт меню *Правка* (Edit) → *Вставка* (Paste).

- Вычислить значения функций $X(\alpha)$ и $Y(\alpha)$ при $\alpha:=\pi/2$.

Этап 8

- Для функции двух переменных $X(t,\alpha):=t\cdot\cos(\alpha)\cdot\sin(\alpha)$ построить график поверхности *Поверхности* (Surface plot) и карту линий уровня *Контурный* (Contour plot) несколькими способами и сравнить их:
  1. По умолчанию, то есть не задавая никаких значений по осям.
  2. С использованием встроенной функции *CreateMesh* (*function, s0, s1, t0, t1, sgrid, tgrid, fmap*). Она возвращает вложенный массив из трех матриц, представляющих координаты *x*, *y* и *z* параметрической поверхности, определенной функцией двух переменных в первом аргументе.
     С помощью этой функции построить график поверхности, при этом сетку задать размером 40×40, диапазон изменения *t* от –5 до 5, α — от 0 до $2\cdot\pi$. Отразить в отчете.
  3. Задав поверхность математически:
     — определить аналогичную исходной функцию $X1\ (t, \alpha)$;
     — задать на осях переменных *t* и α по 41 точке, то есть определить i:=0..40 и j:=0..40. Для переменной t_i со значениями, изменяющимися от –5 до 5, рассчитать требуемый шаг, а для переменной α_j от 0 до $2\cdot\pi$ с шагом $\pi/20$ задать $\alpha_j:=\pi/20\cdot j$;
     — определить матрицу $M_{i,j}:=X1\ (t_i, \alpha_j)$ и отобразить ее графически.
- С помощью команды *Формат* (*Format*) → *Graf* → *3-DPlot* контекстного меню вызвать диалоговое окно *3-D Plot Format* и изменить:

— характеристики просмотра *General* → *View* → *Rotation, Tilt;*
— цвета и линии поверхности *Appearance* → *Line Options, Full Options;*
— параметры осей *Axes;*
— вид заголовка графика *Title*.

Этап 9

- Используя переменную *FRAME* и команду *View*→*Animate*, создать анимационные клипы с помощью данных, приведенных в табл. 22 (для одного из вариантов по указанию преподавателя). Способ создания анимационного клипа следующий: Mathcad имеет встроенную переменную *FRAME*, чье единственное назначение — управление анимациями. Для создания клипа необходимо:
  — создать объект, чей вид зависит от *FRAME*;
  — убедиться, что установлен режим автоматического расчета (*Tools* → *Calculate* → *Automatic Calculation*);
  — выбрать *Вид* (*View*) → *Анимация* (*Animate*) для вызова одноименного диалогового окна (или *Tools* → *Animation* → *Record*);
  — заключить в выделяющий пунктирный прямоугольник часть рабочего документа, которую нужно анимировать;
  — установить нижние и верхние границы *FRAME* (поля *От* (*From*): и *До* (*To*):);
  — в поле Скорость (*At.)* задать значение скорости воспроизведения (кадров/сек);
  — выбрать *Анимация* (*Animate)*. Сейчас анимация только создается;
  — сохранить анимацию как AVI файл (*Сохранить как* (*Save as*)).

11. Решение уравнений и систем

Наиболее интересными, с точки зрения инженерного подхода, являются возможности системы Mathcad по решению уравнений и систем уравнений. Как известно, многие уравнения и системы уравнений не имеют аналитических решений. В первую очередь это относится к большинству трансцендентных уравнений. Доказано также, что нельзя построить формулу, по которой можно было найти решение произвольного алгебраического уравнения, имеющего степень больше четвертой. Однако такие уравнения могут решаться численными методами (итерационными) с заданной точностью (не более значения заданного системной переменной *TOL*).

Численное решение нелинейного уравнения

Для простейших уравнений вида $f(x) = 0$ решение в Mathcad находится с помощью функции *root (f (x), x, a, b)*.

На рис. 87 показан вариант решения одного уравнения при помощи указанной функции. Найдите в справочной системе Mathcad ее описание и самостоятельно сделайте вывод о назначении 3- и 4-го параметров в 3-м способе использования указанной функции (отразить в отчете).

Для успешной работы функции *root* необходимо задать начальное приближение, то есть присвоить первоначальное значение неизвестному параметру. Приближенные значения корней (*начальные приближения*) могут быть известны из физического смысла задачи, из решения аналогичной задачи при других исходных данных или найдены графическим способом.

Таблица 22

№ вар.	Переменные и функции	FRAME	Тип графика
1	$x := 0, 0.1.. 30$ $f(x) := x + \mathrm{FRAME}$	от 0 до 20	График полярные координаты
2	$i := 0..20$ $j := 0..20$ $f(x,y) := = \sin\begin{pmatrix} x^2+y^2 \\ +FRAME \end{pmatrix}$ $x_i := -1.5 + 0.15 \cdot i$ $y_i := -1.5 + 0.15 \cdot j$ $M_{i,j} := f(x_i, y_j)$	от 0 до 50	График поверхности. В метке для ввода матрицы укажите *M*
3	$R := (\mathrm{FRAME}-116)/10$ $t := -11.6, -11.5.. R$ $x(t) = \sin(\sqrt{2} \cdot t) + \cos(2 \cdot t)$ $y(t) = \sqrt{2} \cdot \cos(\sqrt{2} \cdot t) - 2 \cdot \sin(2 \cdot t)$	от 0 до 1000	*X-Y* зависимость. Границы на осях: *x* от –2 до 2; *y* от –5 до 5 показать по осям *y*(*t*), *y*(*R*) и *x*(*t*), *x*(*R*)
4	$\alpha := \pi + \mathrm{FRAME}/5$ $x := 0, 0.1..4 \cdot \pi$ $f(x) := x \cdot \sin(x)$ $f1(x) := \frac{d}{dx} f(x)$ $y(\alpha, x) = f1(\alpha) \cdot (x - \alpha) + f(\alpha)$	от 0 до 60	*X-Y* зависимость. Границы на осях: *x* от 0 до 4π; *y* от –15 до 10 показать по осям *f*(*x*), *y*(α, *x*), *f*(α) с меткой и *x, x,* α
5	$i := 0..\ \mathrm{FRAME} + 1$ $g_i := 0.5 \cdot i \cdot \cos(i)$ $h_i := i \cdot \sin(i)$ $k_i := 2 \cdot i$	от 0 до 50	*3D* точечный график. Границы на осях min/max: *x* –50/50; *y* –50/50; *z* 0/50. В метке для ввода матрицы укажите (*g*, *h*, *k*)
6	$r := \mathrm{FRAME}$ $R = 6$ $n := 0..20$ $m := 0..20$ $v_n := 2 \cdot \pi \cdot n / (r+1)$ $w_m = 2 \cdot \pi \cdot m / (r+1)$ $x_{m,n} = (R + r \cdot \cos(v_n)) \cdot \cos(w_m)$ $y_{m,n} = (R + r \cdot \cos(v_n)) \cdot \sin(w_m)$ $z_{m,n} = r \cdot \sin(v_n)$	от 0 до 20	График поверхности (границы на всех осях установить от –11 до 11). В метке для ввода матрицы укажите (*x, y, z*)

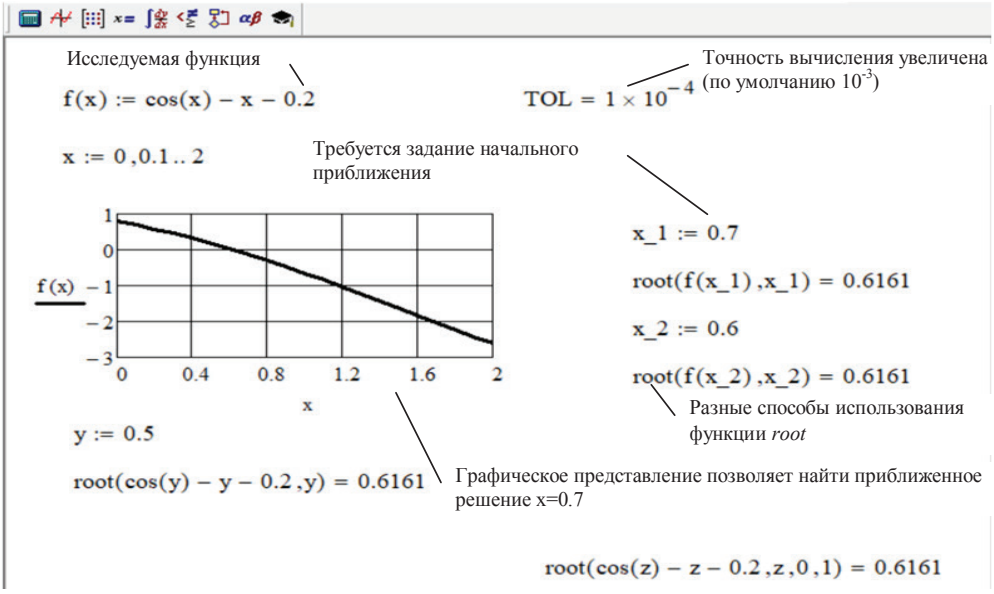


Рис. 87. Решение одного нелинейного уравнения

Наиболее распространен графический способ определения начальных приближений. Принимая во внимание, что действительные корни уравнения $f(x) = 0$ — это точки пересечения графика функции $f(x)$ с осью абсцисс, достаточно построить график функции $f(x)$ и отметить точки пересечения $f(x)$ с осью Ox или отметить на оси Ox отрезки, содержащие по одному корню. Построение графиков часто удается значительно упростить, заменив уравнение $f(x) = 0$ равносильным ему уравнением вида $f_1(x) = f_2(x)$, где функции $f_1(x)$ и $f_2(x)$ — более простые, чем исходная функция $f(x)$. Тогда, построив графики функций $y = f_1(x)$ и $y = f_2(x)$, искомые корни получим как абсциссы точек пересечения этих графиков.

Если после многих итераций Mathcad не находит подходящего приближения, то появится сообщение «Can't converge to a solution» (отсутствует сходимость).

Чтобы установить причину ошибки, исследуйте график $f(x)$. Он поможет выяснить наличие корней уравнения $f(x) = 0$ и, если они есть, определить приблизительно их значения. Чем точнее выбрано начальное приближение корня, тем быстрее будет *root* сходиться.

Этап 1

- Построить график функции $f(x)$ в соответствии с заданным преподавателем вариантом (табл. 23) и приблизительно определить один из корней уравнения (можно использовать команду *Trace* контекстного меню графика). Решить уравнение $f(x) = 0$ с точностью $\varepsilon = 10^{-4}$ с помощью встроенной функции Mathcad *root*.

Таблица 23

№ вар.	Функция, интервал	№ вар.	Функция, интервал
1	$e^{x-1} - x^3 - x,\ x \in [0;2]$	9	$0.25 \cdot x^3 + x - 2,\ x \in [0;2]$
2	$x - \dfrac{1}{3 + \sin(3.6 \cdot x)},\ x \in [0;3]$	10	$a\cos\left(\dfrac{1 - x^2}{1 + x^2}\right) - x,\ x \in [1;4]$
3	$a\cos(x) - \sqrt{1 - 0.3 \cdot x^3},\ x \in [-1;1]$	11	$3 \cdot x - 4 \cdot \ln(x) - 5,\ x \in [2;5]$
4	$3 \cdot x - 14 + e^x - e^{-x},\ x \in [0;3]$	12	$e^x - e^{-x} - 2,\ x \in [0;3]$
5	$\sqrt{1 - 0.4 \cdot x^2} - a\sin(x),\ x \in [-1;3]$	13	$\sqrt{1 - x} - \tan(x),\ x \in [0;2]$
6	$\sqrt{2 \cdot x^2 + 1.2 - \cos(x)} - 1,\ x \in [0;1]$	14	$1 - x + \sin(x) - \ln(1 + x),\ x \in [0;3]$
7	$\cos\left(\dfrac{2}{x}\right) - 2 \cdot \sin\left(\dfrac{1}{x}\right) + \dfrac{1}{x},\ x \in [1;4]$	15	$x^{-2} - x - 0.2,\ x \in [0.5;2]$
8	$0.1 \cdot x^2 - x \cdot \ln(x),\ x \in [0.5;\ 2]$	16	$e^{-x} + \cos(3 \cdot x),\ x \in [1;2]$

Нахождение корней полинома

Для нахождения корней выражения вида $v_n \cdot x^n + \ldots + v_2 \cdot x^2 + v_1 \cdot x + v_0$ лучше использовать функцию *polyroots (v)*, нежели *root*. В отличие от функции *root* функция *polyroots* не требует начального приближения и возвращает сразу все корни как вещественные, так и комплексные.

На рис. 88 показан пример определения корней полинома 3-й степени. Вектор коэффициентов полинома удобно определять, используя пункт меню *Symbolics→Polynomial Coefficients*, предварительно поставив курсор на переменную x в полиноме.

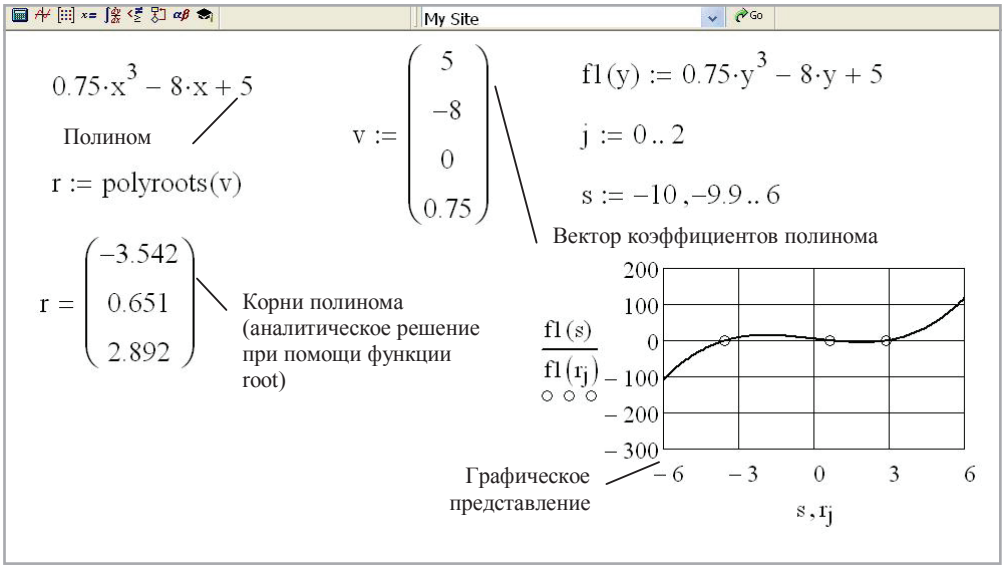


Рис. 88. Нахождение корней полинома

В ряде случаев удобно «объединять» математическое выражение и комментарий к нему, пример использования такой возможности показан на рис. 87, 88. Достичь подобного объединения можно следующим образом: в процессе написания комментария (текстовой области) выбрать пункт меню *Insert* → *Math Region*, при этом появится квадратик черного цвета, означающий, что здесь можно задавать математическое выражение.

Этап 2

- Для варианта полинома $g(x)$, приведенного в табл. 24, выполнить следующие действия:
  1) создать вектор V, содержащий коэффициенты полинома;
  2) решить уравнение $g(x) = 0$ с помощью функции *polyroots,* отобразить корни графически;
  3) решить уравнение символьно, используя команду *Simbolics* → *Variable* → *Solve,* сделать выводы.

Таблица 24

№ вар.	$g(x)$	№ вар.	$g(x)$
1	$x^4-2\cdot x^3+x^2-12\cdot x+20$	9	$x^4+x^3-17\cdot x^2-45\cdot x-100$
2	$x^4+6\cdot x^3+x^2-4\cdot x-60$	10	$x^4-5\cdot x^3+x^2-15\cdot x+50$
3	$x^4-14\cdot x^2-40\cdot x-75$	11	$x^4-4\cdot x^3-2\cdot x^2-20\cdot x+25$
4	$x^4-x^3+x^2-11\cdot x+10$	12	$x^4+5\cdot x^3+7\cdot x^2+7\cdot x-20$
5	$x^4-x^3-29\cdot x^2-71\cdot x-140$	13	$x^4-7\cdot x^3+7\cdot x^2-5\cdot x+100$
6	$x^4+7\cdot x^3+9\cdot x^2+13\cdot x-30$	14	$x^4+10\cdot x^3+36\cdot x^2+70\cdot x+75$
7	$x^4+3\cdot x^3-23\cdot x^2-55\cdot x-150$	15	$x^4+9\cdot x^3+31\cdot x^2+59\cdot x+60$
8	$x^4-6\cdot x^3+4\cdot x^2+10\cdot x+75$	16	$x^4-3\cdot x^3-42\cdot x^2-71\cdot x+14$

Решение систем уравнений

Mathcad дает возможность решать и системы уравнений. Максимальное число уравнений и переменных равно 50. Результатом решения системы будет численное значение искомого корня.

Для решения системы уравнений необходимо выполнить следующее:

— задать начальное приближение для всех неизвестных, входящих в систему уравнений, поскольку Mathcad решает систему с помощью итерационных методов;

— напечатать ключевое слово *Given*. Оно указывает Mathcad, что далее следует система уравнений;

— ввести уравнения и неравенства в любом порядке. Используйте [Ctrl]= для печати символа $\boxed{=}$. Между левыми и правыми частями неравенств может стоять любой из символов <, >, ≥ и ≤;

— введите любое выражение, которое включает функцию *Find*, например: $a := Find(x, y)$.

Ключевое слово *Given*, уравнения и неравенства, которые следуют за ним, и какое-либо выражение, содержащее функцию *Find*, называют блоком решения уравнений. Функция $Find(z1, z2, ...)$ возвращает точное решение системы уравнений. Число аргументов должно быть равно числу неизвестных. Найденное решение можно вывести, напечатав выражение вида $Find(var1, var2,..) =$.

Или задается переменная с помощью функции *Find*. Например, $a := Find(x)$ — скаляр, $var := Find(var1, var2, ...)$ — вектор. Кроме

того, можно определить другую функцию, например $f(a, b, c, ...) := Find(x, y, z, ...)$. Приведенная конструкция удобна для многократного решения системы уравнений для различных значений некоторых параметров a, b, c, ..., непосредственно входящих в систему уравнений.

Рассмотрим систему n линейных алгебраических уравнений относительно n неизвестных $x_1, x_2, ..., x_n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2, \\ \qquad\qquad \cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n. \end{cases}$$

В соответствии с правилом умножения матриц рассмотренная система линейных уравнений записывается в матричном виде

$$Ax = b,$$

где $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$, $b = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{bmatrix}$ — матрица системы, вектор-столбец неизвестных и вектор-столбец правых частей соответственно. Если матрица А — неособенная, то есть $\det A \neq 0$, то система уравнений, или эквивалентное ей матричное уравнение, имеет единственное решение. В самом деле, при условии $\det A \neq 0$ существует обратная матрица A^{-1}. Умножая обе части уравнения на матрицу A^{-1}, получим $A^{-1} \cdot A \cdot x = A^{-1} \cdot b$; $x = A^{-1} \cdot b$.

Приведенная формула дает единственное решение матричного уравнения. В Mathcad системы линейных уравнений удобно решать с помощью функции *lsolve (A, b)*. Пример ее использования приведен на рис. 89.

$ORIGIN := 1$

$x_1 := 0 \quad x_2 := 0 \quad x_3 := 0$

Начальные приближения

Given

$3 \cdot x_1 - x_2 = 5$

$-2 \cdot x_1 + x_2 + x_3 = 0$

$2 \cdot x_1 - x_2 + 4 \cdot x_3 = 15$

$Rez := Find(x_1, x_2, x_3)$

Первый способ использование *Given-Find*

$$Rez = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

$$A := \begin{pmatrix} 3 & -1 & 0 \\ -2 & 1 & 1 \\ 2 & -1 & 4 \end{pmatrix}$$

$$B := \begin{pmatrix} 5 \\ 0 \\ 15 \end{pmatrix}$$

$X1 := A^{-1} \cdot B$

$$X1 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

Второй способ - решение матричного уравнения

$X2 := lsolve(A, B)$

$$X2 = \begin{pmatrix} 2 \\ 1 \\ 3 \end{pmatrix}$$

Использование функции *lsolve*

Given

$a + 2 \cdot \pi \cdot h = q1$

$4 \cdot a + h = m1$

$$Find(a, h) \rightarrow \begin{pmatrix} -\dfrac{q1 - 2 \cdot \pi \cdot m1}{8 \cdot \pi - 1} \\ -\dfrac{m1 - 4 \cdot q1}{8 \cdot \pi - 1} \end{pmatrix}$$

Решение системы уравнений в символьном виде

Рис. 89. Решение системы уравнений

Этап 3

- В соответствии с указанным преподавателем вариантом (табл. 25) решить систему линейных уравнений:
  — с использованием функции *Find*;
  — матричным способом;
  — через функцию *lsolve*.

Символьное решение уравнений и систем уравнений

В Mathcad можно быстро и точно найти численное значение корня с помощью функции *root*. Но имеются некоторые задачи, для которых возможности Mathcad позволяют находить решения в символьном (аналитическом) виде (табл. 26).

Таблица 25

№ вар.	Система линейных уравнений	№ вар.	Система линейных уравнений
1	$\begin{cases} 2\cdot x_1+x_2+2\cdot x_3+3\cdot x_4=8 \\ 3\cdot x_1+3\cdot x_3=6 \\ 2\cdot x_1-x_2+3\cdot x_4=4 \\ x_1+2\cdot x_2-x_3+2\cdot x_4=4 \end{cases}$	9	$\begin{cases} 2\cdot x_1+x_2-5\cdot x_3+x_4=-4 \\ x_1-3\cdot x_2-6\cdot x_4=-7 \\ 2\cdot x_2-x_3+2\cdot x_4=2 \\ x_1+4\cdot x_2-7\cdot x_3+6\cdot x_4=-2 \end{cases}$
2	$\begin{cases} x_1+2\cdot x_2+3\cdot x_3+4\cdot x_4=22 \\ 2\cdot x_1+3\cdot x_2+x_3+2\cdot x_4=17 \\ x_1+x_2+x_3-x_4=8 \\ x_1-2\cdot x_3-3\cdot x_4=-7 \end{cases}$	10	$\begin{cases} x_1+2\cdot x_2+3\cdot x_3+4\cdot x_4=29 \\ 2\cdot x_1+3\cdot x_2-4\cdot x_3+x_4=20 \\ 3\cdot x_1+4\cdot x_2+x_3-x_4=16 \\ 4\cdot x_1+x_2+2\cdot x_3+3\cdot x_4=19 \end{cases}$
3	$\begin{cases} 9\cdot x_1+10\cdot x_2-7\cdot x_3-x_4=23 \\ 7\cdot x_1-x_3-5\cdot x_4=7 \\ 5\cdot x_1-2\cdot x_3+x_4=22 \\ 4\cdot x_1+x_2+2\cdot x_3+3\cdot x_4=26 \end{cases}$	11	$\begin{cases} 2\cdot x_1-8\cdot x_2-3\cdot x_3-2\cdot x_4=-18 \\ 2\cdot x_1-2\cdot x_2+3\cdot x_3-2\cdot x_4=28 \\ x_2+x_3+x_4=10 \\ 11\cdot x_2+x_3+2\cdot x_4=21 \end{cases}$
4	$\begin{cases} 6\cdot x_1-x_2+10\cdot x_3-x_4=158 \\ 2\cdot x_1+x_2+11\cdot x_3+7\cdot x_4=120 \\ 3\cdot x_1-2\cdot x_2-2\cdot x_3-x_4=7 \\ x_1-12\cdot x_2+2\cdot x_3-x_4=17 \end{cases}$	12	$\begin{cases} 2\cdot x_1-x_2+4\cdot x_3+x_4=66 \\ 2\cdot x_2-6\cdot x_3+x_4=-7 \\ 8\cdot x_1-3\cdot x_2+6\cdot x_3-5\cdot x_4=97 \\ 2\cdot x_1-7\cdot x_2+6\cdot x_3-x_4=54 \end{cases}$
5	$\begin{cases} x_1-2\cdot x_2+6\cdot x_3+x_4=88 \\ 5\cdot x_1+2\cdot x_3-3\cdot x_4=50 \\ 7\cdot x_1-3\cdot x_2+7\cdot x_3+2\cdot x_4=117 \\ 3\cdot x_1-7\cdot x_2+5\cdot x_3+2\cdot x_4=99 \end{cases}$	13	$\begin{cases} 2\cdot x_1-3\cdot x_3-2\cdot x_4=-16 \\ 2\cdot x_1-x_2+13\cdot x_3+4\cdot x_4=73 \\ 3\cdot x_1+x_2+2\cdot x_3+x_4=126 \\ x_1-12\cdot x_2-5\cdot x_4=-26 \end{cases}$
6	$\begin{cases} x_1-2\cdot x_2-8\cdot x_4=-7 \\ x_1+4\cdot x_2-7\cdot x_3+6\cdot x_4=-8 \\ x_1+x_2-5\cdot x_3+x_4=-10 \\ 2\cdot x_1-2\cdot x_2+2\cdot x_4=7 \end{cases}$	14	$\begin{cases} 7\cdot x_1+7\cdot x_2-7\cdot x_3-2\cdot x_4=5 \\ 3\cdot x_1+4\cdot x_2+5\cdot x_3+8\cdot x_4=60 \\ 2\cdot x_1+2\cdot x_2+2\cdot x_3+x_4=47 \\ 2\cdot x_1-2\cdot x_3-x_4=-1 \end{cases}$
7	$\begin{cases} 2\cdot x_1+2\cdot x_2+6\cdot x_3+x_4=15 \\ -x_2+2\cdot x_3+x_4=18 \\ 4\cdot x_1-3\cdot x_2+x_3-5\cdot x_4=37 \\ 3\cdot x_1-5\cdot x_2+x_3-x_4=30 \end{cases}$	15	$\begin{cases} 6\cdot x_1-9\cdot x_2+5\cdot x_3+x_4=118 \\ 7\cdot x_2-5\cdot x_3-7\cdot x_4=-14 \\ 5\cdot x_1-3\cdot x_2+2\cdot x_3+x_4=61 \\ 3\cdot x_1-9\cdot x_2+x_3+6\cdot x_4=45 \end{cases}$
8	$\begin{cases} 4\cdot x_1-5\cdot x_2+7\cdot x_3+5\cdot x_4=165 \\ 2\cdot x_1+x_2-3\cdot x_3-x_4=-15 \\ 9\cdot x_1+4\cdot x_3-x_4=194 \\ x_1-x_2-2\cdot x_3-3\cdot x_4=-17 \end{cases}$	16	$\begin{cases} x_1-2\cdot x_2-8\cdot x_4=-7 \\ x_1+4\cdot x_2-7\cdot x_3+6\cdot x_4=12 \\ x_1+x_2-5\cdot x_3+x_4=27 \\ 2\cdot x_1-x_2+2\cdot x_4=7 \end{cases}$

Решение уравнений в символьном виде позволяет найти точные или приближенные корни уравнения. Команда *Simbolics* → *Variable* → *Solve* позволяет решить уравнение относительно некоторой переменной и выразить его корни через остальные параметры уравнения.

Чтобы решить уравнение символьно необходимо:

— напечатать выражение (для ввода знака равенства используйте комбинацию клавиш [Ctrl]=);

— выделить переменную, относительно которой нужно решить уравнение, щелкнув на ней мышью;

— выбрать пункт меню *Simbolics* → *Variable* → *Solve*.

Чтобы решить систему уравнений в символьном виде, необходимо выполнить следующее:

— напечатать ключевое слово *Given*;

— напечатать уравнения в любом порядке ниже слова *Given*. Удостоверьтесь, что для ввода знака = используется [Ctrl]=;

— напечатать функцию *Find*, соответствующую системе уравнений;

— Нажать [Ctrl]. (клавиша CTRL, сопровождаемая точкой). Mathcad отобразит символьный знак равенства →.

— Щелкнуть мышью на функции *Find*.

Этап 4

Символьно решить системы уравнений, приведенные в табл. 26. Внимание! Если имена переменных были использованы ранее (в предыдущих заданиях), сделать замену.

Таблица 26

Вариант	Задание 1	Задание 2
1	$\begin{cases} 3\cdot x+4\cdot\pi\cdot y=a \\ 2\cdot x+y=b \end{cases}$	$\begin{cases} 2\cdot y-\pi\cdot z=a \\ \pi\cdot z-y=b \\ 3\cdot x+y=c \end{cases}$
2	$\begin{cases} 2\cdot\pi\cdot x-2\cdot y=a \\ x+4\cdot y=b \end{cases}$	$\begin{cases} 3\cdot\pi\cdot x-2\cdot y=a \\ y+\pi\cdot z=2\cdot b \\ 5\cdot x+\pi\cdot y=c \end{cases}$
3	$\begin{cases} 6\cdot x-2\cdot\pi\cdot y=s \\ x-4\cdot y=f \end{cases}$	$\begin{cases} x-2\cdot\pi\cdot y+z=w \\ 4\cdot y-z+\pi=t \\ x+12\cdot y=v \end{cases}$

Окончание табл. 26

Вариант	Задание 1	Задание 2
4	$\begin{cases} 3\cdot\pi\cdot x+y-1=z \\ x+2\cdot\pi\cdot y=h \end{cases}$	$\begin{cases} 2\cdot x+y-3\cdot\pi\cdot z=f \\ x-1.5\cdot\pi\cdot y+z=g \\ 2\cdot y-4\cdot z=k \end{cases}$
5	$\begin{cases} 4\cdot x-3\cdot\pi\cdot y=2\cdot d \\ 2\cdot\pi\cdot x+y=f \end{cases}$	$\begin{cases} x+2\cdot\pi\cdot y=a \\ 3\cdot y-\pi\cdot z=2\cdot b \\ x+\pi\cdot y+z=c \end{cases}$
6	$\begin{cases} \pi\cdot x-3\cdot y=2\cdot\pi\cdot m \\ 4\cdot x+y=n \end{cases}$	$\begin{cases} 2\cdot x-\pi\cdot y+z=w \\ 4\cdot x-\pi\cdot z=q \\ 2\cdot y+3\cdot z=t \end{cases}$

Приближенные решения

Очень часто бывает, что система уравнений просто не имеет решения или же задача заключается в поиске минимального значения некоторых величин, в этом случае применяется функция *Minerr()*, которая использует алгоритм функции *Find*. Если в результате поиска не может быть получено дальнейшее уточнение текущего приближения к решению, *Minerr* возвращает это приближение. Другими словами, возвращаемое значение — вектор таких значений аргументов, которые приводят к минимальной ошибке в системе уравнений и неравенств, начинающихся от ключевого слова *Given*. Правила использования функции *Minerr* такие же, как и функции *Find*. Если *Minerr* используется в блоке решения уравнений, необходимо всегда включать дополнительную проверку достоверности результатов.

Этап 5

- Для заданного преподавателем варианта (табл. 27) построить графики каждого из нелинейных уравнений системы и определить начальное приближение решения (для этого преобразовать уравнения к виду $f_1(x) = y$ и $f_2(y) = x$). Решить систему нелинейных уравнений с помощью функции *Minerr*.

Этап 6

- Используя возможности Mathcad, повторить вычисления, проведенные в разделе «Работа с числовыми матрицами в программе Excel», включая индивидуальное задание. Сделать вывод.

Таблица 27

№ вар.	Система нелинейных уравнений	№ вар.	Система нелинейных уравнений
1	$\begin{cases}\sin(x)+2\cdot y=2\\ \cos(y-1)+x=0.7\end{cases}$	9	$\begin{cases}\sin(y)+x=-0.4\\ 2\cdot y-\cos(x+1)=0\end{cases}$
2	$\begin{cases}\sin(x+0.5)-y=1\\ \cos(y-2)+x=0\end{cases}$	10	$\begin{cases}\sin(x+2)-y=1.5\\ \cos(y-2)+x=0.5\end{cases}$
3	$\begin{cases}\cos(x)+y=1.5\\ 2\cdot x-\sin(y-0.5)=1\end{cases}$	11	$\begin{cases}\cos(x+0.5)-y=2\\ \sin(2\cdot y)-x=1\end{cases}$
4	$\begin{cases}\cos(x+0.5)+y=0.8\\ \sin(y)-2\cdot x=1.6\end{cases}$	12	$\begin{cases}\cos(2-x)+y=0\\ \sin(y+0.5)-x=1\end{cases}$
5	$\begin{cases}\sin(x-1)=1.3-y\\ x-\sin(y+1)=0.8\end{cases}$	13	$\begin{cases}\cos(x+0.5)+3\cdot y=1\\ \sin(y+0.5)-x=0.7\end{cases}$
6	$\begin{cases}\cos(x+0.5)+y=3\\ \sin(y)-2\cdot x=2\end{cases}$	14	$\begin{cases}\sin(x)-2\cdot y=1.5\\ \cos(y+0.5)-2\cdot x=4\end{cases}$
7	$\begin{cases}-\sin(x+1)+y=0.8\\ \sin(y-1)+x=1.5\end{cases}$	15	$\begin{cases}2\cdot y-\sin(x-1.5)=1\\ \cos(y+0.2)+x=2.5\end{cases}$
8	$\begin{cases}\sin(x)-2\cdot y=1\\ \sin(y-1)+x=1.3\end{cases}$	16	$\begin{cases}\sin(2\cdot x-0.5)-y=1\\ \cos(2-y)+x=0\end{cases}$

12. Решение оптимизационных задач. Интегрирование и дифференцирование средствами Mathcad

Функция *Find* поиска решения систем уравнений имеет ряд недостатков, а именно: чувствительность к начальному приближению, необходимость присутствия ключевого слова *Given*, отсутствие сходимости в случае, если функция является плоской в окрестности точки решения.

Как альтернативу можно использовать функцию *Minerr ()*, пример которой был рассмотрен в работе 2, и функции *Maximize (f, var1, var2,..)* и *Minimize (f, var1, var2,..)*. Они возвращают значения *var1*, *var2*, ..., которые «заставляют» функцию *f* принять свое максимальное (минимальное) значение. Максимизация (минимизация) может проводиться с ограничениями, которые записываются между ключевым словом *Given* и функцией *Maximize* (*Minimize*). Вышеперечисленные функции удобно использовать для решения оптимизационных задач, широко распространенных в энергетике. Далее рассмотрим две типовые оптимизационные задачи.

Задача 1. Пусть некий завод электротехнического оборудования может выпускать трансформаторы двух типов, стоимостью в 165 и 110 условных единиц. Основные затраты заключаются в расходе меди (медная обмотка), стали (стальной сердечник) и прочих материалов. Известно количество запасов материалов на заводском складе (табл. 28). Спрашивается — каков должен быть план выпуска трансформаторов для обеспечения заводу максимальной прибыли.

Таблица 28

Трансформатор	Расход стали, усл. т	Расход меди, усл. т	Прочие материалы, усл. т
Тип № 1	85	2	7
Тип № 2	50	3	4
Ресурс	3000	110	340

Неизвестные: x_1 — число трансформаторов 1-го типа и x_2 — число трансформаторов 2-го типа. На рис. 90 графически представлены ограничения, задаваемые ресурсами затрачиваемых материалов. Дополнительно — x_1 и x_2 не отрицательны и не могут быть дробным числом (кому надо полтора трансформатора?). Решение задачи в среде Mathcad показано на рис. 91.

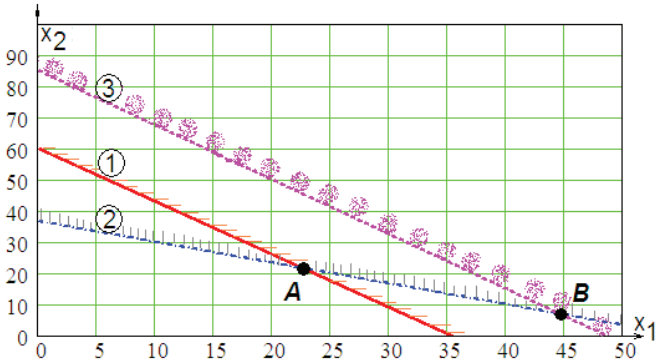


Рис. 90. Графическое представление ограничений

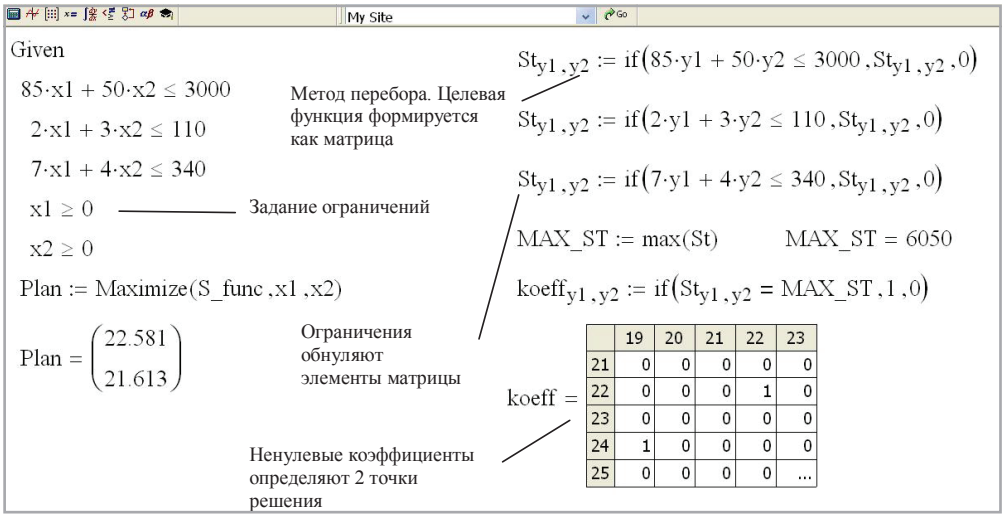


Рис. 91. Решение задачи в среде Mathcad двумя способами

Самостоятельно вы можете убедиться, что решение, предложенное Mathcad, соответствует точке А на рис. 90. К сожалению, Mathcad не умеет решать задачу целочисленного линейного программирования, поэтому предложенный им вариант не является удовлетвори-

тельным. «Обойти» указанную проблему можно тривиальным методом перебора (рис. 91).

На данном рисунке приведен пример использования логического оператора *if (*cond, *x, y)*. Вложенная функция *if* («если») возвращает *x*, если условие cond верно (истинно, ненулевое) и в противном случае возвращает *y*.

Еще одна функция — max (*A, b*) возвращает максимальное значение из *A*, *b*, причем количество аргументов функции может быть любым (больше одного), аргументами могут быть матрица, вектор, скаляр.

Данная задача имеет два решения: максимальной прибыли соответствует выработка 24-х трансформаторов 1-го типа и 19 трансформаторов 2-го. Либо по 22 трансформатора обоих типов.

Второй тип оптимизационных задач — уже известная задача линейного программирования так называемая транспортная задача.

На рис. 92 приведено решение в среде Mathcad задачи линейного программирования, подробно рассмотренной в разделе «Решение транспортной задачи в программе Excel».

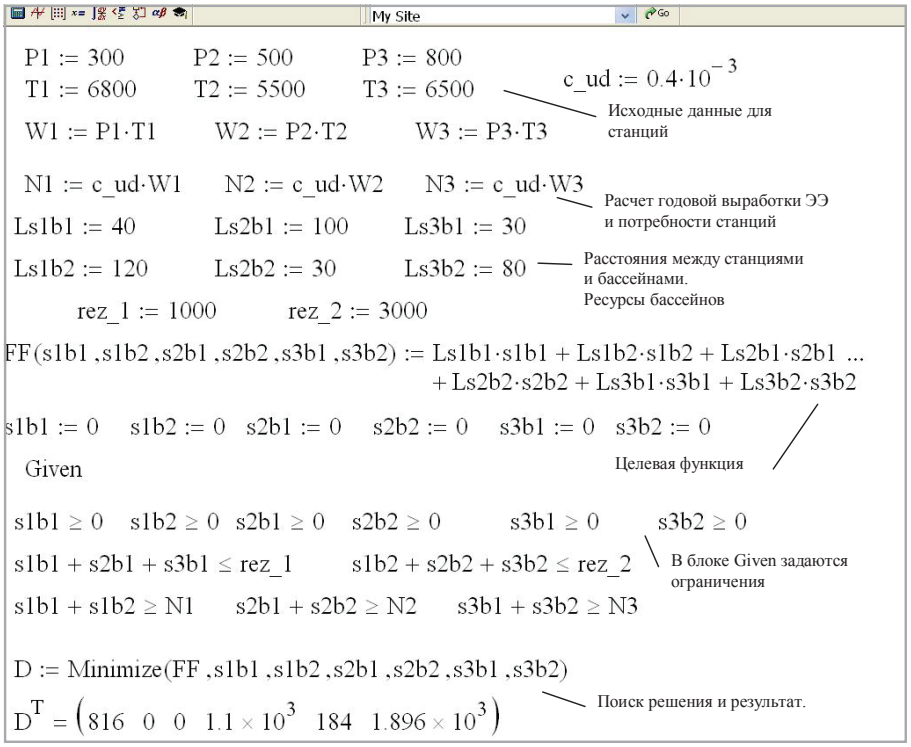


Рис. 92. Транспортная задача

Функции *Minimize* и *Maximize* позволяют находить экстремум функции, нелинейно зависящей от неизвестных параметров, с заданными ограничениями в виде равенств и неравенств. Но в этом случае всегда надо выполнять проверку на наличие дополнительных экстремумов, так как нелинейные целевые функции часто многоэкстремальны. Решение в этом случае зависит от выбора начального приближения.

Этап 1

- Повторить решение «задачи о трансформаторах», приведенное на рис. 91.
- Выполнить в среде Mathcad индивидуальное задание — решение транспортной задачи, сравнить результаты с полученными в среде Excel. Сделать вывод.

Интегрирование и дифференцирование

Процессы, протекающие в электроэнергетических сетях, описываются системами нелинейных уравнений, решение которых требует навыков в выполнении интегрирования, дифференцирования, аппроксимации, умения применять численные и итерационные методы.

Решение дифференциального уравнения (системы) можно выполнять различными методами. Наиболее распространен метод Эйлера, но справочная система Mathcad дает информацию, что в случае использования конструкции *d/d* дифференцирование выполняется методом Ридерса (Ridder`s). Кроме того, к услугам пользователя Mathcad предлагает еще 14 функций решения систем дифференциальных уравнений. Нередко приходится пробовать на одном дифференциальном уравнении (одной системе) несколько методов, чтобы определить, какой метод лучше (быстрее, точнее). Некоторые возможности символьной математики (или компьютерной математики аналитических преобразований) уже были рассмотрены. Но в ряде случаев Mathcad становится бессилен и надеяться целиком на него в нахождении аналитических выражений все же не стоит. Примеры применения символьных вычислений приведены на рис. 93 и 94.

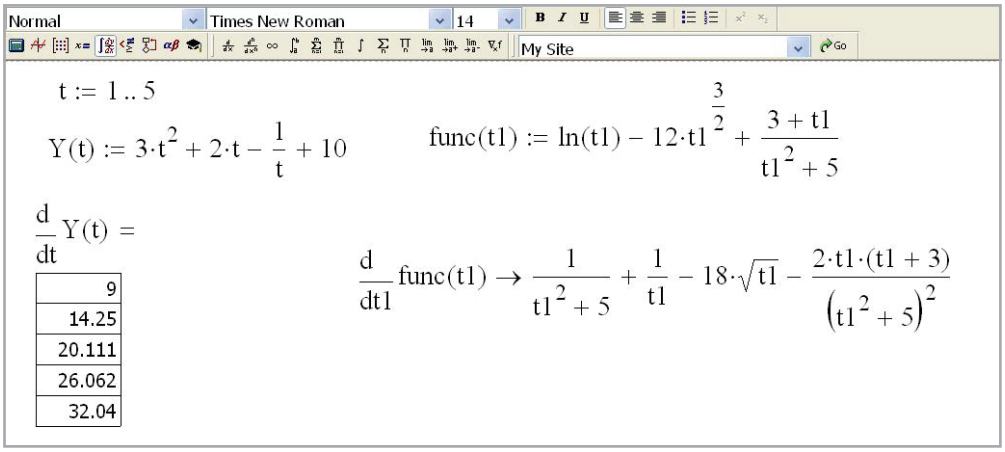


Рис. 93. Пример дифференцирования

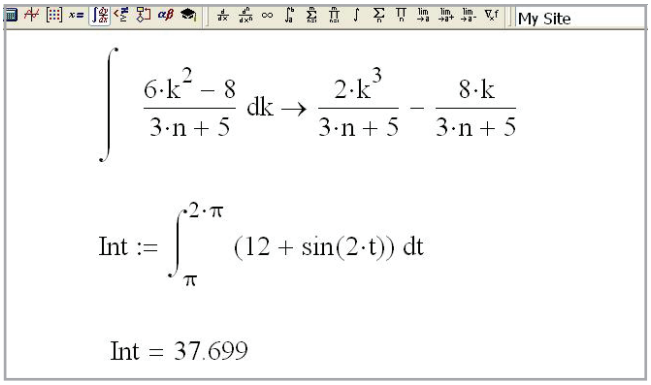


Рис. 94. Пример интегрирования

Этап 2

В соответствии с заданным преподавателем вариантом (табл. 29) вычислить предложенные значения численно и символически.

Таблица 29

№ п/п	Определить первую и вторую производную функций. Определить числовое значение на интервале [a, b], число шагов 10.	Определить первообразную и вычислить интеграл на интервале [a, b]
1	$7x^{15}+9x^4+5x+8$ [0, 1]; $\sin(2x)+x^4$ [1, 4]	x^5+6x^2 [1, 9]; $5\sin(x)-1/(x+20)$ [0, 5]
2	$3x+10x^2$ [1, 2]; $\sin(x)\sqrt{x+8}$ [0, 4]	$1/x^2+e^x-5(x-3)^3$ [1, 6]; $\cos(2x+3)-1/(x+3)^2$ [3, 8]
3	$\sqrt[3]{x^4+6x^3-3}$ [−1, 2]; $\sin(2x+5)-\tan(x)$ [1, 2]	$x^{3/2}+1/\sin(x)$ [1, 5]; $x^{2/3}-3x^3+12x+1/x^2$ [5, 9]
4	$(x+12)/(x^2+4x+3)$ [0, 5]; $4x-\sin(2x^2-3x+4)$ [0, 4]	$1/(x^2-4x+6)$ [1, 6]; $x-2+1/(x^3+4)$ [3, 7]
5	$15\ln(x)/(x^3+2x-1)$ [1, 3]; $\tan 3x-1/(6+x)$ [3, 6]	$\tan(x)/\cos(x+\pi)+\sin(2x)$ [0, π]; $(x+4)/(x^{1/2}+3)$ [0, 5]
6	$\ln(1+4x)+3(x^2-4)^2$ [0, 2] $e^{2x}-14(1+x^{2/3})\ln(x+3)$ [0, 4]	$x^2-\ln(x)^2+e^{x-2}$ [2, 6]; $(4x-3)^{3/2}+(x-5)^2/x$ [1, 5]

13. Основы работы в Access

Системы управления базами данных (СУБД), к категории которых относится и приложение Access, были разработаны прежде всего для эффективной работы с «плоскими» или «двумерными» таблицами. В отличие от табличного процессора Excel, СУБД представляют гораздо большие возможности на всех стадиях работы с данными и не ограничены жесткими рамками допустимого объема информации [6, 7].

База данных Access состоит из следующих основных элементов: таблицы (основной элемент), запросы, формы, макросы, модули, отчеты.

В ходе выполнения «Заданий» предоставляется возможность последовательно разобраться в назначении и особенностях перечисленных элементов, создать свои экземпляры этих элементов, научиться управлять ими. Итог занятия — учебная база данных, которая в несколько упрощенном варианте повторяет черты реальной.

Этап 1. Импорт таблицы

Запустите Access, выберите команду *Создать* и, задав имя «База по учету электроэнергии», сохраните новую базу в вашей личной папке на рабочем диске.

Для того чтобы быстрее приступить к работе, используем постановку задачи и рабочую книгу «Ведомость учета электроэнергии», которая является результатом вводного раздела по Excel. Напомним, что для некоторой «фирмы», осуществляющей отпуск электроэнергии потребителям (Иванову, Петрову, Сидорову), была запрограммирована таблица, в которой на основе данных о расходе электроэнергии конкретными потребителями и сумм, поступивших в качестве оплаты, рассчитывалась задолженность абонентов.

Таким образом, первоочередная задача — осуществить преобразование разработанных в Excel структур в базу данных. Для этого с помощью закладки *Внешние данные* командой *Импорт электронной таблицы Excel* откройте рабочую книгу под названием «Ведомость учета электроэнергии», расположенную на сетевом диске (можно использовать и свою книгу, оставшуюся от занятий на Excel, отменив в ней подведение итогов и сортировку). Выполняется это в режиме *Импортировать данные источника в новую таблицу в текущей базе данных*. Не рекомендуется при импортировании таблицы формировать «ключевое поле» — лучше это сделать позже. После следования рекомендациям *мастера импорта* в вашей базе данных появится первая таблица «Лист1: таблица» (рис. 95).

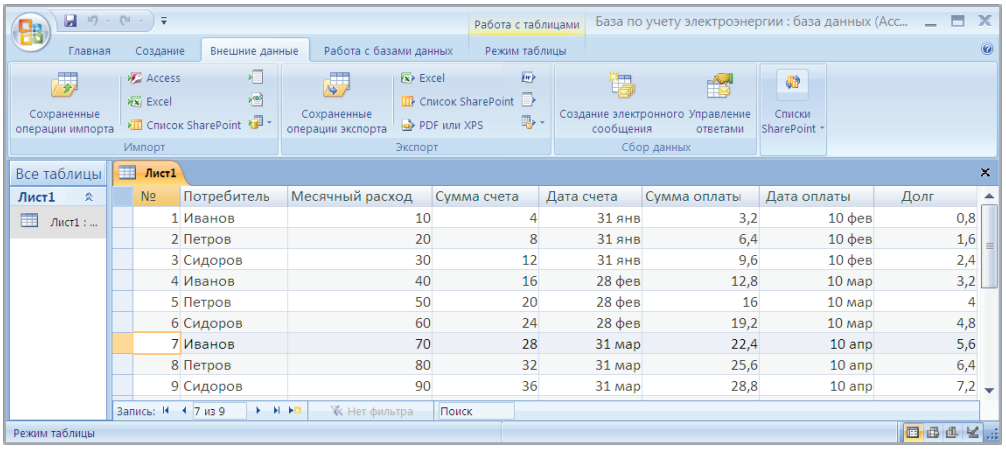


Рис. 95. Первый этап создания таблицы *Основной лист*

В импортированной таблице можно удалить столбцы «Сумма счета» и «Долг». В полученной таким образом таблице остались только исходные (первичные) данные, а все результаты будут рассчитаны средствами Access. Поле «№» переименуйте в «Код» и исправьте по следующему принципу: код Иванова — 1, Петрова –2, Сидорова — 3. Также замените названия столбцов, сделав их стандартными для подобных таблиц. Так «Потребитель» заменяется на «Фамилия», «Месячный расход» на «Количество электроэнергии» и «Дата счета» на «Дата замера» (рис. 96). Если по какой-либо причине импорт указанной рабочей книги Excel невозможен, создайте по образцу рисунка таблицу средствами Access. Для этого на закладке *Созда-*

ние кнопкой *Таблица* запустите создание новой таблицы в режиме *Конструктора* (рис. 97).

Лист1

Код	Фамилия	Количество электроэнергии	Дата замера	Сумма оплаты	Дата оплаты
1	Иванов	10	31 янв	3,2	10 фев
2	Петров	20	31 янв	6,4	10 фев
3	Сидоров	30	31 янв	9,6	10 фев
1	Иванов	40	28 фев	12,8	10 мар
2	Петров	50	28 фев	16	10 мар
3	Сидоров	60	28 фев	19,2	10 мар
1	Иванов	70	31 мар	22,4	10 апр
2	Петров	80	31 мар	25,6	10 апр
3	Сидоров	90	31 мар	28,8	10 апр

Рис. 96. Второй этап создания таблицы

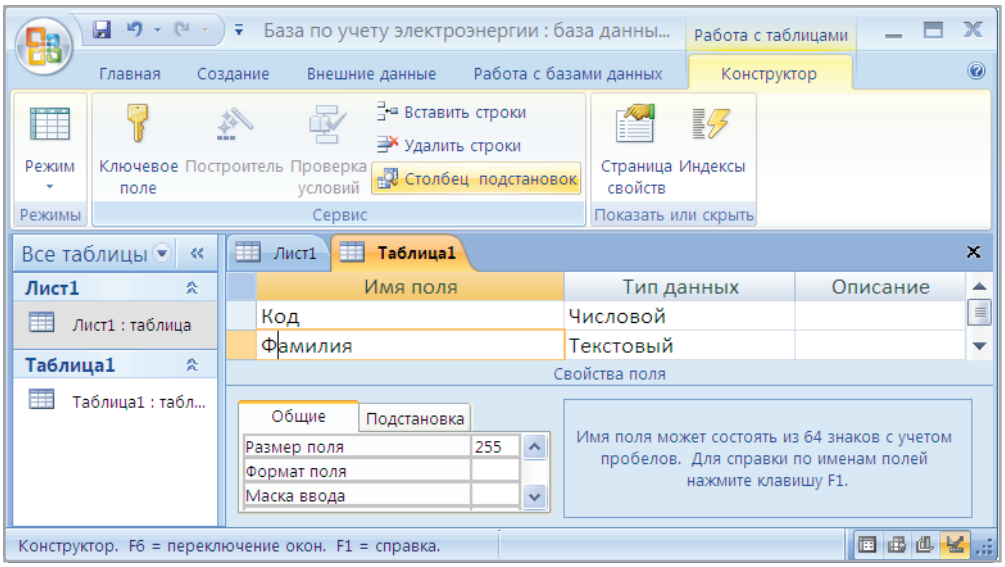


Рис. 97. Создание таблицы в режиме конструктора

Этап 2. Разделение импортированной таблицы

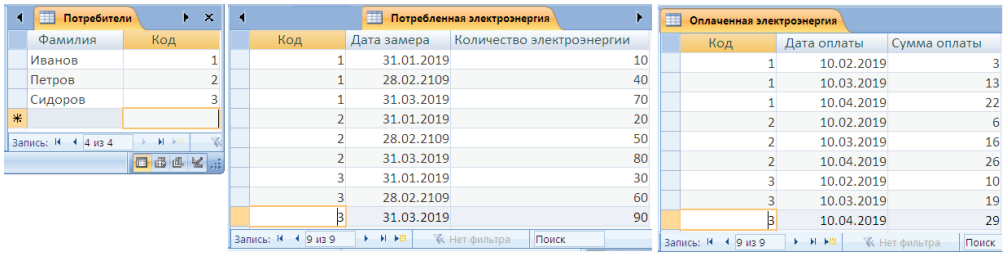
Таблица «Лист 1» имеет неоптимальную структуру, о чем свидетельствуют множественные повторения одних и тех же данных в различных записях.

Для начальной оптимизации «первичных» данных эту таблицу следует разбить на три.

Разбивку таблицы «Основной лист» можно выполнить различными способами: копированием или специальными запросами.

Разбивка таблицы копированием

Для этого в режиме *Конструктора* таблиц по очереди сформируйте три новые таблицы (рис. 98). При этом после заполнения имен полей (названия также можно получить поэлементным копированием из импортированной таблицы «Лист1» в режиме переименования) и открытия записей (хотя бы нулевыми данными в каком-либо одном поле) можно блоками перекопировать содержимое соответствующих полей. Чтобы отменить признак ключевого поля, достаточно отжать соответствующую кнопку на ленте *Конструктора*. Состав базы на данный момент должен соответствовать рис. 99.



Потребители

Фамилия	Код
Иванов	1
Петров	2
Сидоров	3

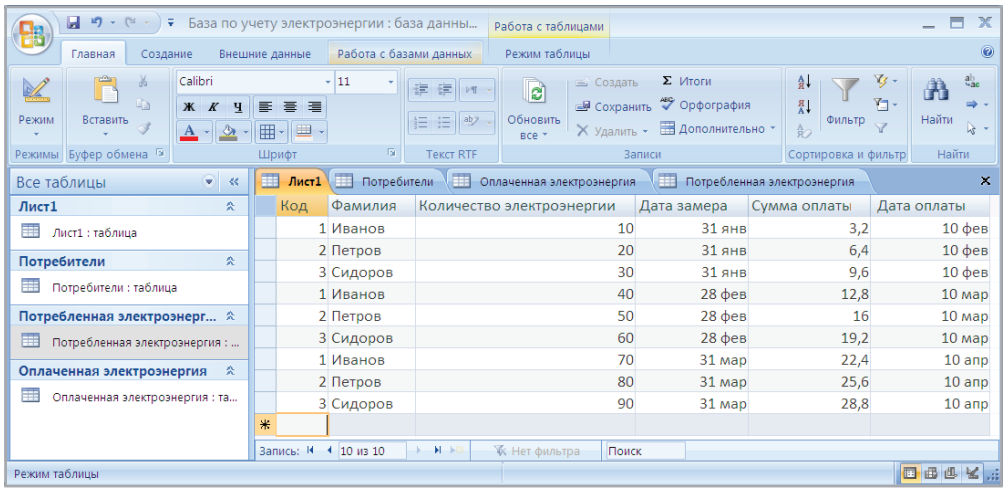
Потребленная электроэнергия

Код	Дата замера	Количество электроэнергии
1	31.01.2019	10
1	28.02.2109	40
1	31.03.2019	70
2	31.01.2019	20
2	28.02.2109	50
2	31.03.2019	80
3	31.01.2019	30
3	28.02.2109	60
3	31.03.2019	90

Оплаченная электроэнергия

Код	Дата оплаты	Сумма оплаты
1	10.02.2019	3
1	10.03.2019	13
1	10.04.2019	22
2	10.02.2019	6
2	10.03.2019	16
2	10.04.2019	26
3	10.02.2019	10
3	10.03.2019	19
3	10.04.2019	29

Рис. 98. Результат разбивки таблицы «Лист1»



Лист1

Код	Фамилия	Количество электроэнергии	Дата замера	Сумма оплаты	Дата оплаты
1	Иванов	10	31 янв	3,2	10 фев
2	Петров	20	31 янв	6,4	10 фев
3	Сидоров	30	31 янв	9,6	10 фев
1	Иванов	40	28 фев	12,8	10 мар
2	Петров	50	28 фев	16	10 мар
3	Сидоров	60	28 фев	19,2	10 мар
1	Иванов	70	31 мар	22,4	10 апр
2	Петров	80	31 мар	25,6	10 апр
3	Сидоров	90	31 мар	28,8	10 апр

Рис. 99. Состав базы данных

Разбивка таблицы с помощью запросов

Запрос — это специальный объект базы данных, обеспечивающий заданную обработку таблиц. Чаще всего такой обработкой является выбор данных по какому-либо критерию (с фильтрацией, сортировкой, вычислениями и т. п.).

Access предоставляет возможности наглядного (визуального) программирования запроса, но на самом деле любой запрос — это конечный набор инструкций специального языка для управления базами данных — SQL (structured query language).

Первый шаг вполне стандартен: в главном окне базы данных выбирается закладка *Создание* и выполняется команда *Конструктор запросов*. В автоматически всплываемом окне *Добавление таблицы* в качестве источника выберите таблицу «Лист1», а из нее два поля: «Код» и «Потребитель». Последнее выполняется перетаскиванием названия соответствующего поля в нижнюю часть окна, которое называется *бланком запроса*. Если бланк запроса отсутствует, раздвиньте его горизонтальную границу (рис. 100). Для того, чтобы результатом выполнения запроса стала новая таблица, на закладке конструктора запроса выполните команду *Создание таблицы*, задав соответствующее имя. Сама таблица будет сформирована после выполнения запроса с помощью нажатия на кнопку с «жирным» восклицательным знаком на той же закладке.

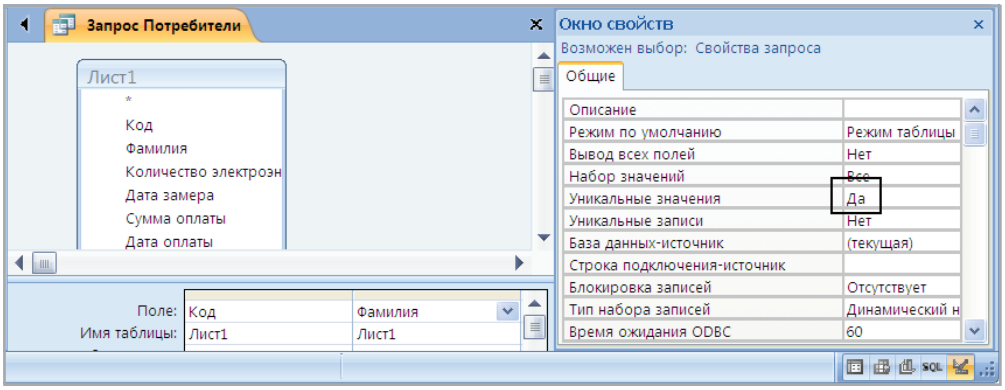


Рис. 100. Окно создания запроса

Первый вариант таблицы «Потребители» не может быть назван удовлетворительным (каждая запись повторяется в ней трижды!). После включения режима *уникальных записей* в окне свойств запроса (рис. 100) Access выполнит необходимое объединение записей.

Аналогичным образом с помощью отдельных запросов формируются и две оставшиеся таблицы — «Потребленная электроэнергия» и «Оплаченная электроэнергия». В них объединение выполнять не нужно. Запросы на создание таблиц можно не сохранять — их действие разовое.

Решение практических задач, как правило, требует объединения данных, расположенных в отдельных таблицах. Для этого между полями таблиц устанавливаются специальные отношения, называемые связями. Главное требование: тип и формат данных связующих полей должен быть одинаков.

По типу связи бывают «один ко многим» (обозначается «1-----∞») и «многие ко многим» (обозначается «∞-----∞»). Второй тип связей используется чаще всего при «механическом» разделении большой таблицы на части, ориентированные, к примеру, на конкретных пользователей, а вот первый тип и является основным для связывания отдельных таблиц. Таблица, в которой связанное поле является ключевым (сторона «один»), называется родительской, а таблица со стороны «многие» — дочерней. Связующее поле в дочерней таблице называется «внешним» или «чужим» ключом.

Для подготовки к связыванию исходных таблиц поле «Код» таблицы «Потребители» сделайте ключевым. Выполняется это командой *Ключевое поле* в группе *Сервис* в режиме *Конструктор таблицы* или в нижней части окна *Конструктора таблицы* на закладке *Общие* в свойстве *Индексированное поле* значением *Да* (*Совпадения не допускаются*). Затем командой *Схема данных* на закладке *Работа с базами данных* установите связи один ко многим между таблицами (рис. 101). Сама связь устанавливается универсальным приемом *взял и тащи*, применяемому к полю *Код* таблицы *Потребители*. Связь не установится, если тип и формат полей *Код* в связываемых таблицах различаются. Рекомендуется установить их числовыми и целыми.

При установке связи между таблицами в окне Связи устанавливаются параметры объединения:

- *обеспечение целостности данных* — установка данного параметра не позволит, например, в таблицу «Оплаченная электроэнергия» внести данные с кодом несуществующего потребителя или удалить из таблицы «Потребители» запись для абонента, присутствующего в связанных таблицах «Потребленная электроэнергия», «Оплаченная электроэнергия»;

- *каскадное обновление данных* — данный параметр позволяет автоматически изменять значения внешних ключей при изменениях в родительской таблице;
- *каскадное удаление связанных полей* — предусматривает удаление соответствующих записей в дочерних таблицах.

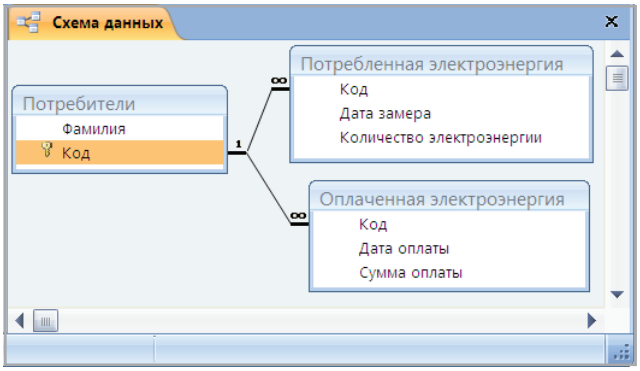


Рис. 101. Задание связей

Этап 3. Формирование запросов

Итоговую обработку данных, представленных в базе, сосредоточим в запросах. Для большей наглядности создадим три запроса. С помощью первого запроса определяется суммарное количество электроэнергии, отпущенное конкретному потребителю. Запрос практически не увеличивает размер базы данных. Проектирование запроса возможно в трех режимах: инструкциями SQL, с помощью мастера или с помощью конструктора. Последний способ был уже частично освоен на этапе 2, поэтому работа будет продолжена в том же режиме.

Запрос «Потребление: итого»

После включения режима *Создание запроса в режиме конструктора* необходимо добавить таблицы — источники данных: «Потребители» и «Потребленная электроэнергия», а затем выбрать необходимые поля: «Фамилия» и «Количество электроэнергии» (рис. 102).

Пока выполнение данного запроса не дает ничего нового (рис. 103). Первая задача — рассчитать суммарный расход электроэнергии. Для этого на закладке *Конструктора* необходимо нажать кнопку *Итоги* (в бланке запроса появится строка *Групповая операция*), а в поле *Коли-*

чество электроэнергии параметр *Группировка* заменить на *Sum* — операцию суммирования. Проверьте действие запроса. Его результатом должна быть таблица из трех записей с суммарным расходом электроэнергии по каждому потребителю.

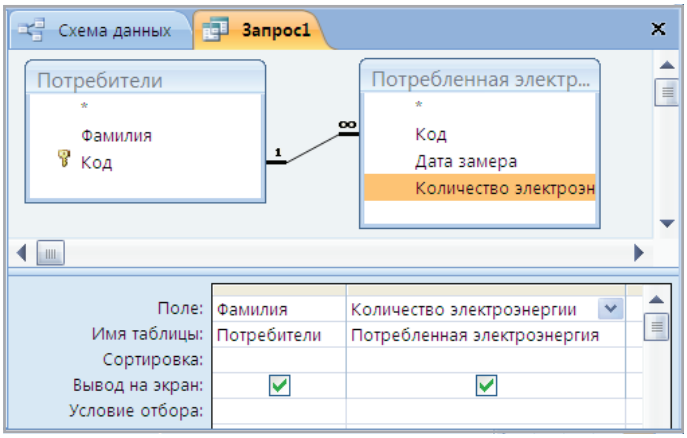


Рис. 102. Создание запроса в режиме конструктора

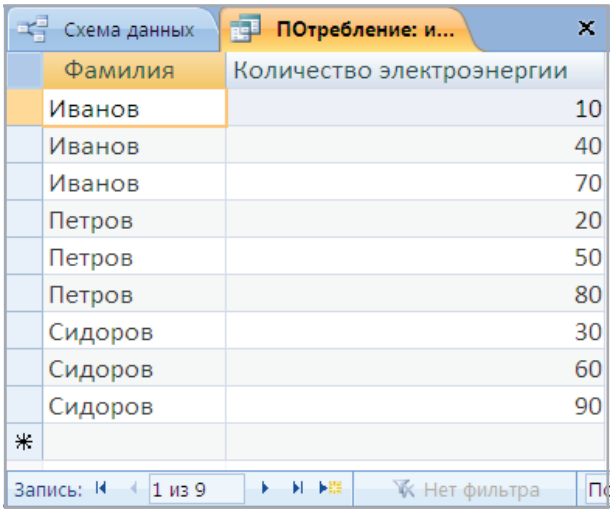


Рис. 103. Результат запроса

Рационально в полученной таким образом таблице сформировать еще один столбец, в котором рассчитать стоимость потребленной электроэнергии. Имя столбца «Стоимость» можно ввести в третий столбец запроса (в режиме конструктора).

Задание расчетной формулы проще выполнить с помощью *построителя выражений*. Для запуска этого мастера курсор устанавливается в поле «Стоимость» и нажимается [Ctrl]F2 (рис. 104).

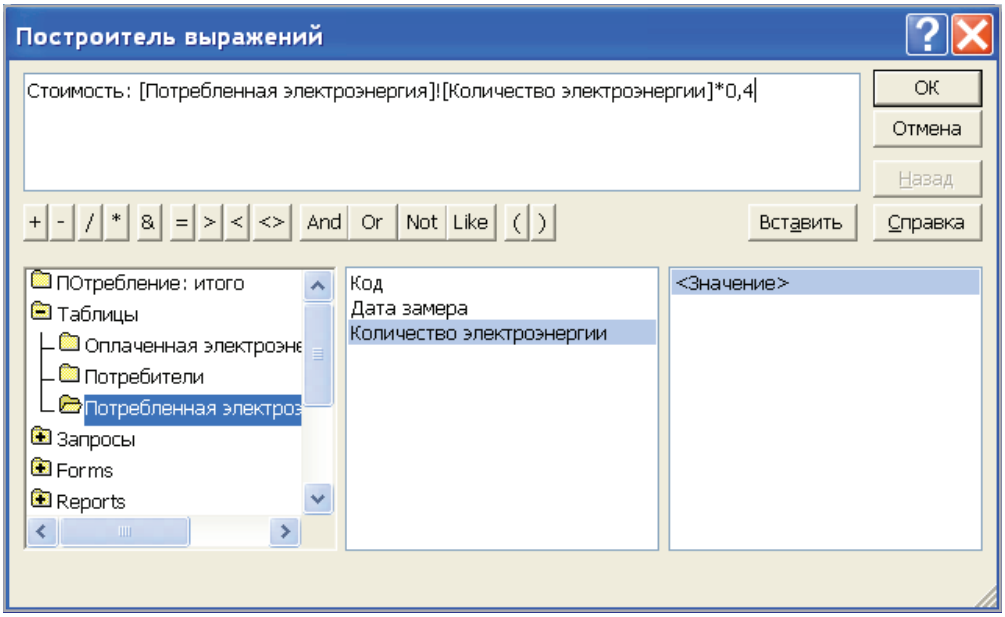


Рис. 104. Построитель выражений

В формулах знак двоеточия означает присвоение результата. В левом нижнем окне построителя выражений присутствует список объектов базы данных, которые могут быть использованы в качестве источников в формулах. Следует открыть группу «Таблицы», выбрать таблицу «Потребленная электроэнергия», а в ней поле — «Количество электроэнергии» (в среднем окне). Знак «!» означает операцию «ссылка», операцию умножения можно выбрать на панели, ну а константу 0,4 (цену киловатт-часа) ввести вручную.

Результатом выполнения запроса в данном случае будет снова таблица из 9 записей, так как в графе *Групповые операции* в поле «Стоимость» конструктор запросов поставил значение «по умолчанию» — Группировка. Поскольку объединение записей по значениям в первом и третьем полях невозможно (нет повторяющихся значений сразу в двух полях) — группировка по сути дела не выполняется. Исправляется эта ситуация просто. В *Групповых операциях* поля «Стоимость» нужно поставить *Sum* — выбрав операцию суммирования значений в груп-

пируемых записях. Результат выполнения запроса должен быть как на рис. 105. После этого запрос можно закрыть с именем «Потребление: итого».

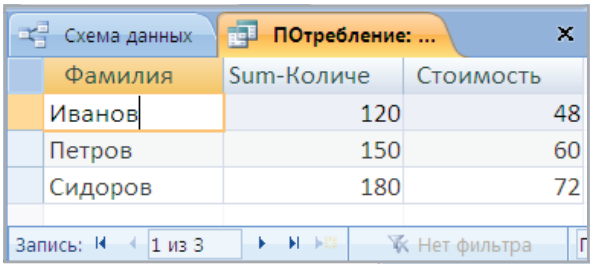


Рис. 105. Результат запроса «Потребление: итого»

Запрос «Оплачено: итого»

В данном запросе не требуется дополнительного поля с вычислениями, а во всем остальном он создается по аналогии с предыдущим запросом. Источником в данном случае являются таблицы «Потребители» и «Оплаченная электроэнергия», а результат выполнения запроса должен соответствовать показанному на рис. 106. Сохраните данный запрос под именем «Оплачено: итого».

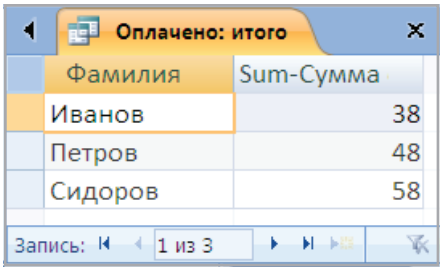


Рис. 106. Запрос «Оплачено: итого»

Запрос «Задолженность»

Отличием данного запроса от предыдущих является использование в качестве источников также запросов «Потребление: итого» и «Оплачено: итого». В окне конструктора запроса определяются источники данных и формируются необходимые поля (рис. 107).

Между полями «Фамилия» запросов установите вручную связь, а в поле «Задолженность» с помощью построителя выражений запишите формулу:

Задолженность: [Потребление: итого]! [Стоимость] –

– [Оплачено: итого]! [Sum-Сумма оплаты]

Результат выполнения запроса должен соответствовать рис. 108.

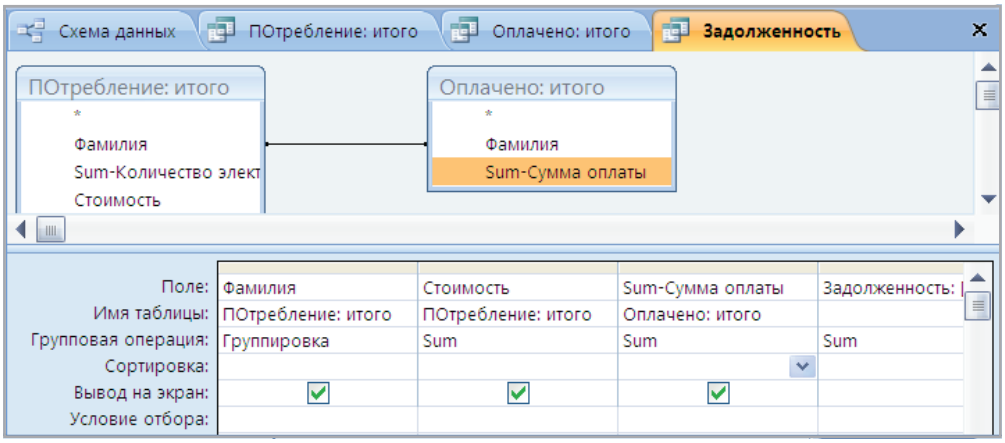


Рис. 107. Формирование запроса «Задолженность»

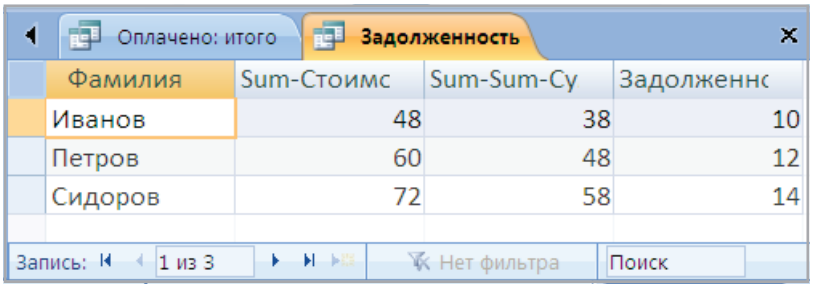


Рис. 108. Запрос «Задолженность»

Сохраните этот запрос под именем «Задолженность». В подготовленных запросах использовалась только одна функция из категории итоговых — суммирования. Кроме нее, при подведении итогов в группируемых записях Access предлагает использовать еще восемь следующих функций:

- *Avg* — среднеарифметическое значение;
- *Min* — наименьшее значение;
- *Max* — наибольшее значение;
- *Count* — количество ненулевых записей;
- *StDev* — стандартное отклонение значений поля;

- *Var* — дисперсия значений поля;
- *First* — первое значение поля;
- *Last* — последнее значение поля.

Этап 4. Разработка форм для ввода данных

Основное назначение формы в Access — обеспечить удобный режим ввода данных в базу. Тем не менее, формы могут быть использованы также и для вывода данных, их коррекции, управления макросами или процедурами, вывода предупреждающих или аварийных сообщений и т. п.

Простая форма для ввода новых потребителей

Первые шаги по проектированию формы подобны тем, что выполнялись при создании запросов: на закладке *Создание* запускается *Конструктор форм*. Окно конструктора в данном случае имеет достаточно своеобразный вид и новые элементы (рис. 109).

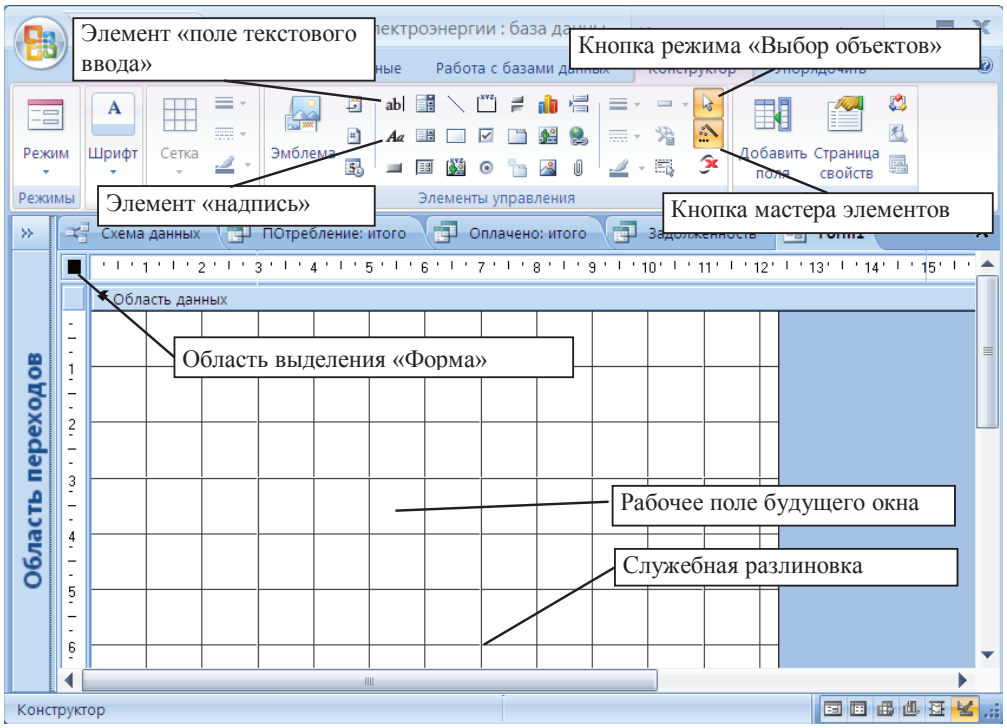


Рис. 109. Окно конструктора формы

Форма — это «проект» окна, которое в своем составе может иметь обычные в Windows-стандарте элементы управления (кнопки, переключатели, поля текстового ввода и пр.). В свою очередь, каждый элемент, размещенный на форме, имеет свой набор параметров или свойств. Проектирование формы в простейшем случае сводится к выбору требуемых элементов управления, размещению их на форме и установлению свойств этих элементов.

Создаваемая форма будет использована для ввода данных в таблицу «Потребители». Поэтому первый шаг ее проектирования — определение указанной таблицы в качестве источника (собственно, и приемника) данных. Свойство *Источник записей* присутствует в окне свойств формы на закладке *Общие*.

Окно свойств формы активизируется двойным щелчком на области выделения формы (квадратик в верхнем левом углу, где начинаются линейки конструктора формы). Того же результата можно достичь и командой *Свойства* из контекстного меню формы (правой клавишей мышки, когда ее указатель находится на области выделения формы).

При помещении курсора в поле свойства *Источник записей* справа появляются две кнопки: списка и построителя запроса. В данном случае необходимо нажать на кнопку списка и в нем выбрать (по имени) таблицу «Потребители». После этого необходимо нажать на кнопку *Добавить поля* в группе *Сервис* на закладке *Конструктора*, в результате чего на рабочем столе появляется дополнительное окно, в котором перечисляются все поля таблицы. Из этого окна либо по отдельности, либо вместе (выделив щелчком первое поле, а затем, удерживая *Shift*, — все последующие поля) следует перетащить поля «Код» и «Фамилия» на проектируемую форму. «Бросать» поля на форму лучше на расстоянии 2,5–3 см от левого края (ориентир — горизонтальная линейка). Дело в том, что каждый раз слева от элемента управления на форме располагается присоединенная надпись. Первоначально она совпадает с именем поля (рис. 110).

На этом же рисунке указаны маркеры, с помощью которых можно изменить размеры элементов формы и их расположение. Добейтесь следующего вида (рис. 111).

Режим перемещения элемента индицируется указателем в виде руки с направленными вверх пальцами.

Осталось проверить форму в действии. Для этого можно выбрать *Режим формы* из контекстного меню формы (правая кнопка мышки

на области выделения формы) или соответствующей командой меню *Вид*. Попробуйте добавить в базу новых потребителей. Например, Семенов с кодом 4, Андреев с кодом 5 и т. п. Закройте форму под именем «Клиенты» и проверьте изменения в таблице «Потребители».

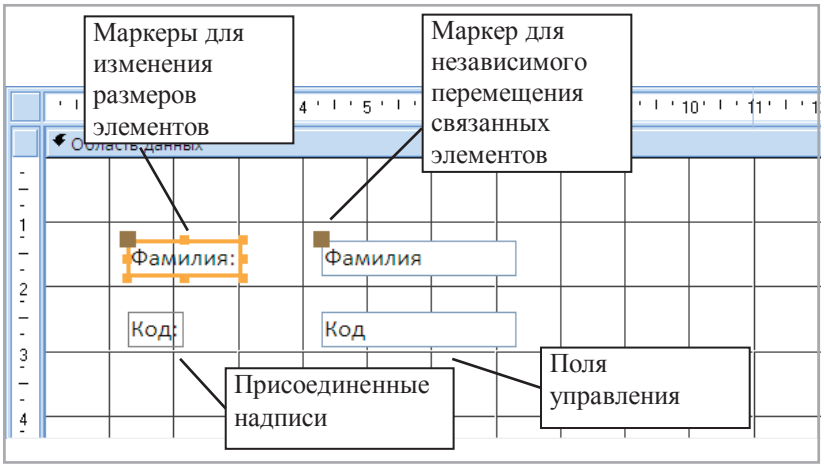


Рис. 110. Расположение присоединенных надписей

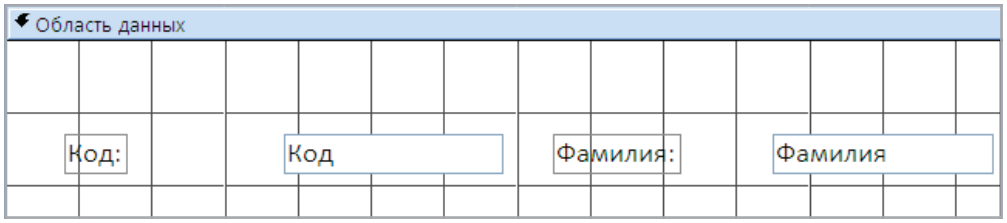


Рис. 111. Разработка формы, первый этап

Вновь откройте форму «Клиенты» в режиме конструктора. Экспериментальным путем установите смысл таких свойств, как *Подпись*, *Область выделения*, *Кнопки перехода*, *Тип границы*, *Кнопки*... Ознакомьтесь с режимом *Ленточная форма*, *Таблица*. Новое значение свойств формы вступает в силу после закрытия окна свойств.

Следующий шаг — добавление на форму элемента. Пусть таковым будет *Надпись*. В режиме конструктора нажимается кнопка *Надпись* и курсом мышки на *Области данных* формы растягивается зона размещения элемента. После отпускания кнопки мышки автоматически включается режим ввода текста. Допустим, это будет пояснение для пользователей формы «Введите код и фамилию потребителя электроэнергии».

После необходимых манипуляций свойствами элементов формы добейтесь вида как на рис. 112.

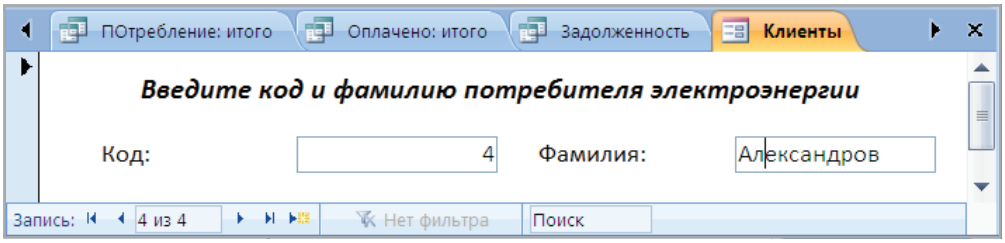


Рис. 112. Форма с надписью

С помощью формы можно проверить в действии параметры объединения таблиц. Формой «Клиенты» замените код Иванова, например на 11, и проверьте, как при этом изменились таблицы «Потребленная электроэнергия» и «Оплаченная электроэнергия». Очевидно, что результаты выполнения запросов должны остаться прежними. Проверьте это. Верните Иванову прежний код, удалите из базы Семенова и Андреева.

Улучшенная форма для таблицы «Потребители»

В создаваемой базе данных кодом потребителя является его порядковый номер. Поэтому можно ограничить объем вводимой информации в таблицу «Потребители» только фамилиями абонентов, а присваивать очередной код будет сам Access.

Для этого в таблицу «Потребители» придется добавить новый столбец, например с именем «Идентификатор». Целесообразнее делать это в режиме конструктора таблиц. Сразу же типом данных нового поля установите *Счетчик*.

Проверьте: значения в полях «Код» и «Идентификатор» должны совпадать. Следующим шагом необходимо поле «Идентификатор» сделать связанным для таблиц базы данных. Сначала удаляются все существующие связи по полю «Код» (меню *Сервис* команда *Схема данных*). Затем с поля «Код» в таблице «Потребители» снимается статус ключевого поля — ключевым полем необходимо сделать «Идентификатор». После чего связи между таблицами устанавливаются вновь, но со стороны таблицы «Потребители» в них участвует поле «Идентификатор».

Проверьте: выполненные в базе преобразования не должны изменить содержание таблиц (кроме таблицы «Потребители») и результаты выполнения запросов.

Удалите с формы элемент «Код», добавьте элемент «Идентификатор», источник записи — соответствующее поле таблицы «Потребители». Так как элемент формы «Идентификатор» связан с полем — счетчиком таблицы, при попытке его изменения в строке сообщений появляется предупредительная надпись. Для исключения этой ситуации свойство *Доступ* поля «Идентификатор» установите в значение *Нет* или свойство *Блокировка* в значение *Да*. Проверьте, и оставьте, на ваш взгляд, более предпочтительный вариант. Измените надпись на форме (уберите упоминание о необходимости введения кода) и добавьте с помощью формы «Клиенты» в базу данных новых абонентов с фамилиями Николаев, Сергеев, Александров.

Простая форма для ввода показаний электросчетчиков

Назначением этой формы будет ввод в базу новых данных по расходованию электроэнергии абонентами. Включите режим конструктора формы и в качестве источника записей выберите на этот раз таблицу «Потребленная электроэнергия». Перенесите из этой таблицы на создаваемую форму поля «Дата замера» и «Количество электроэнергии». Поскольку расход электроэнергии вполне персонифицирован, на форме должно быть поле для задания фамилии потребителя. Однако такого поля в таблице «Потребленная электроэнергия» нет. Есть поле «Код», но заставлять будущего пользователя базы заполнять именно это поле в добавляемой записи не вполне корректно.

Все проблемы решает добавление на форму элемента *поле со списком*. С одной стороны, оно позволит пользователю выбирать фамилии из уже имеющегося реестра (таблица «Потребители»), а во-вторых, вместо фамилии в поле «Код» таблицы «Потребленная электроэнергия» помещать значение идентификатора абонента.

На панели элементов нажимается кнопка *Поле со списком*, после перетаскивания которой на форму открывается окно диалога *Создание полей со списком*. На первом окне следует оставить радио-переключатель в положении Объект поле со списком будет использовать значения из таблицы или запроса. Во втором окне следует выбрать таблицу «Потребители». В третьем окне кнопкой > в подокно Выбранные поля перенести поле «Фамилия». В четвертом окне настраивается сортировка или выбирается ее отсутствие. В пятом окне производится настройка ширины «поля со списком» и скрывается ключевое поле. В шестом окне диалога необходимо выбрать сохранение значения поля со списком в столбце «Код».

Последнее окно мастера по созданию полей со списком позволяет определить присоединенную надпись элемента (например, «Абонент»).

Установите свойство *Ввод данных* формы в значение *Да* и попробуйте с ее помощью ввести замер для потребителя Николаев, выполненный, к примеру, 20.01.2019 и составляющий 25 кВт·час. Сохраните форму под именем «Показания электросчетчиков».

Улучшенная форма для ввода показаний электросчетчиков

Access имеет специальные средства, упрощающие заполнение базы данных. Во-первых, это маски ввода. Для поля «Дата замера» формы «Показания электросчетчиков» в режиме конструктора для *Маски ввода* выберите вариант «Краткий формат даты». В данном случае рекомендуется оставить все установки, предлагаемые *Мастером*. В итоге будет сформирована маска из трех параметров, разделенных символом «;». Первый (обязательный) параметр обеспечивает обязательное заполнение даты цифрами (символ 0), разделение поля на три зоны (разделитель — точки) и направление заполнения справа налево (по умолчанию). Второй и третий параметры (необязательные) предписывают соответственно сохранение маски в базе и использование в качестве заполнителя символа нижнего подчеркивания. Проверьте «маску» в действии.

Имеется возможность еще упростить ввод даты замера с помощью встроенной функции *Date ()* в свойстве *Значение по умолчанию* поля «Дата замера». Функция *Date ()* возвращает текущую дату в кратком формате (дд.мм.гггг). Выполняется это с помощью уже использованного ранее *Построителя выражений*.

Вводимые в базу данные имеет смысл контролировать на допустимые значения. Так, например, дата замера не может быть больше, чем текущая. Дополнительным условием может быть глубина ретроспективы, например полугодие. В окончательном виде свойства поля «Дата замера» формы принимают вид, соответствующий рис. 113.

Показания счетчика не могут быть как отрицательные, так и превосходящие некоторый предел, допустим, в 1000 кВт·час на одного потребителя в месяц. Данное условие реализуется в виде выражения в свойстве поля *Количество электроэнергии* (рис. 114). Остальные свойства поля настраиваются в соответствии с рисунком. В данном случае цифра «9» в маске вода определяет любую необязательную цифру. Строка «квт·час» выводится в поле без изменений и относится к символам маски. Третий параметр маски в данном случае задает пробел в качестве заполнителя.

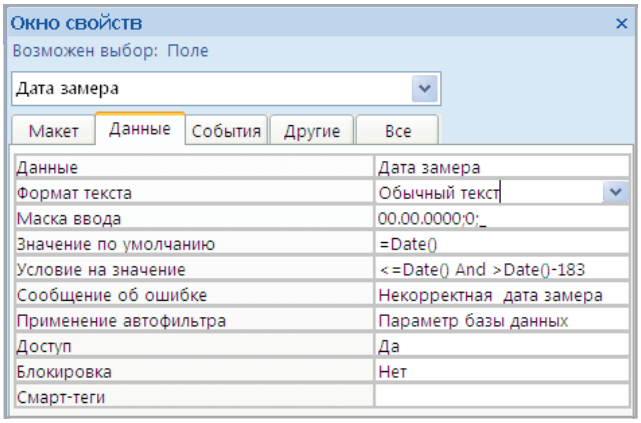


Рис. 113. Свойства поля *Дата замера*

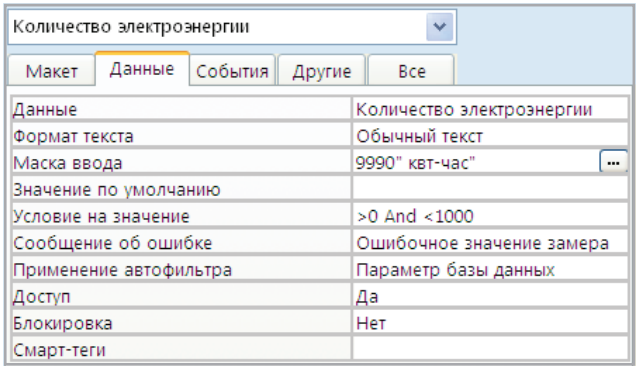


Рис. 114. Свойства поля *Количество электроэнергии*

В окончательном виде форма приобретает вид: рис. 115. Выполните с ее помощью добавление в базу нескольких записей.

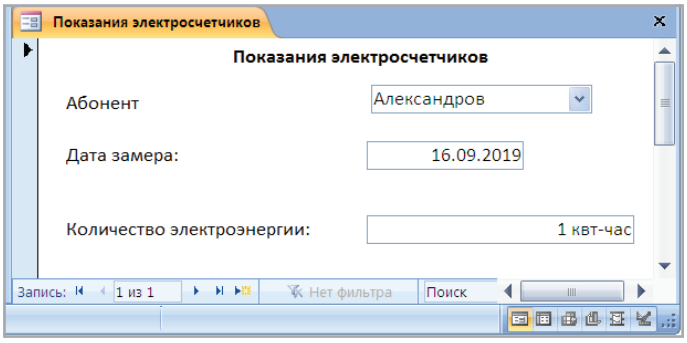


Рис. 115. Форма *Показания счетчиков*

Форма для ввода данных об оплате за электроэнергию

На данной форме необходимо разместить четыре элемента:

— поле со списком, из которого будут выбираться фамилии потребителей;
— поле для задания даты платежа;
— поле для ввода оплаченной суммы;
— надпись, поясняющую назначение формы.

Поле со списком формируется по тому же алгоритму, который был опробован в предыдущей форме. Так, в поле *Дата оплаты* задается маска ввода, значение по умолчанию, допустимое значение и текст сообщения при нарушении заданного условия (рис. 116). В поле для ввода *суммы оплаты* задаются свойства маски, условие на значение и текст сообщения как на рис. 117.

Дата оплаты

Макет | Данные | События | Другие | Все

Данные	Дата оплаты
Формат текста	Обычный текст
Маска ввода	00.00.0000;0;_
Значение по умолчанию	=Date()
Условие на значение	<=Date() And >Date()-183
Сообщение об ошибке	Некорректная дата замера
Применение автофильтра	Параметр базы данных
Доступ	Да
Блокировка	Нет
Смарт-теги	

Рис. 116. Свойства поля *Дата оплаты*

Сумма оплаты

Макет | Данные | События | Другие | Все

Данные	Сумма оплаты
Формат текста	Обычный текст
Маска ввода	9999" руб";0;" "
Значение по умолчанию	
Условие на значение	>0
Сообщение об ошибке	Ошибочное значение платежа
Применение автофильтра	Параметр базы данных
Доступ	Да
Блокировка	Нет
Смарт-теги	

Рис. 117. Свойства поля *Сумма оплаты*

В итоге форма приобретает следующий вид: рис. 118. Выполните с ее помощью добавление в базу нескольких записей.

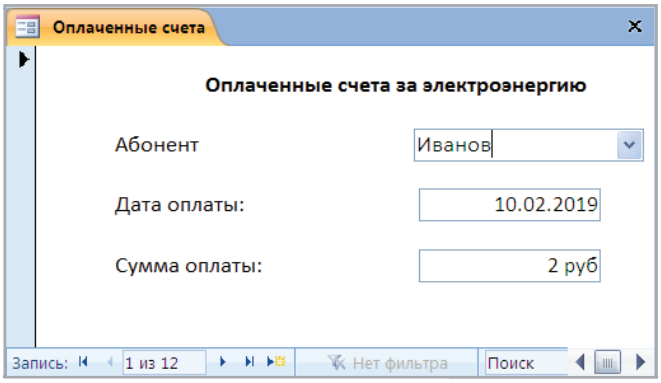


Рис. 118. Форма *Оплаченные счета*

Этап 5. Разработка кнопочной формы

Для придания базе данных *товарного* вида необходимо поработать над ее интерфейсом. Наиболее простым элементом управления в Windows-приложениях являются кнопки. Access умеет автоматически генерировать кнопки для вызова так называемых макросов. Макрос — это объект базы данных, выполняющий предварительно заданную последовательность действий. Действий предусмотрено около 40, выбираются они из раскрывающегося списка.

Задание первого макроса

Первым создается макрос для открытия формы «Клиенты». Для этого на закладке *Создание* главного окна базы данных в группе *Другие* выполняется команда *Макрос*. В окне запустившегося при этом конструктора макросов (рис. 119) в поле *Макрокоманда* выбирается операция, в окне аргументов задается, какая именно форма должна быть открыта. После выполнения соответствующих установок сохраните макрос под именем «МакросКлиенты».

Разработка макросов для остальных объектов базы

Аналогичным образом задаются еще два макроса: «МакросСчетчики» и «МакросСчета», соответственно открывающие формы «Показания счетчиков» и «Оплаченные счета».

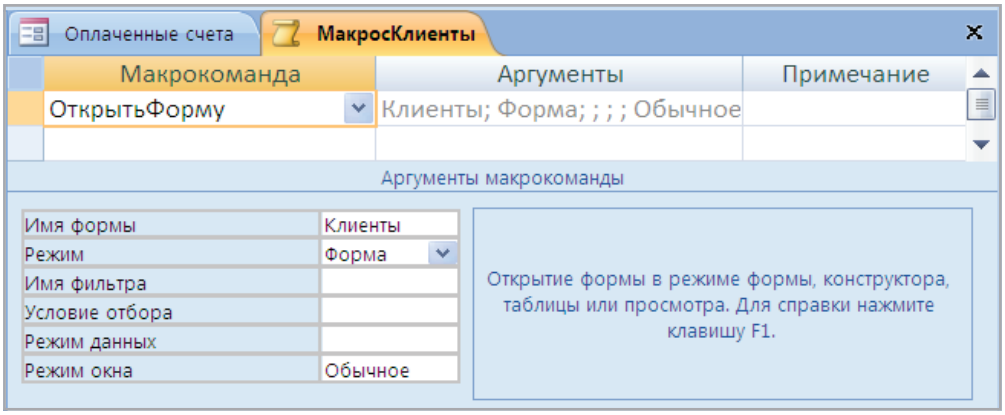


Рис. 119. Конструктор макросов

Для того чтобы пользователю разрабатываемой базы данных можно было контролировать наполнение основных таблиц, необходимы три макроса, каждый из которых открывает соответствующую таблицу: «МакросТаблПотребители», «МакросТаблЭлЭнергия», «МакросТаблОплата». Макрокомандой в данном случае служит *ОткрытьТаблицу*, а аргументами следует запретить ручную правку таблицы (рис. 120).

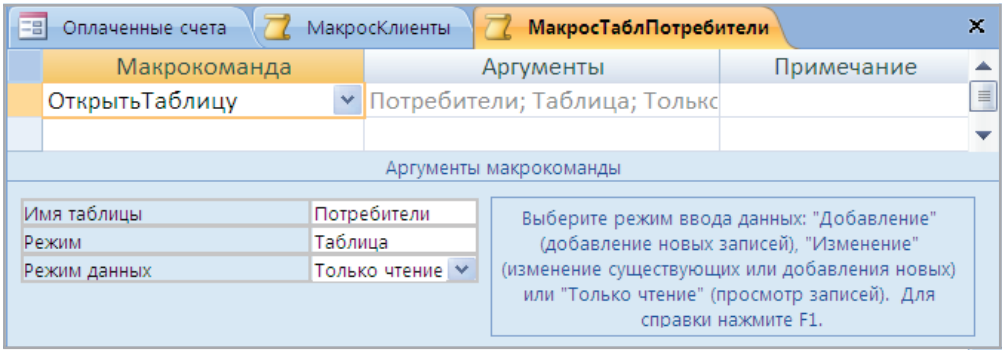


Рис. 120. Макрос для открытия таблицы

По схожему принципу создаются и макросы для запуска запросов. Назовите их, например: *МакросЗапросПотребление*, *МакросЗапросОплата*, *МакросЗапросЗадолженность*. Отдельный макрос *МакросВыход*, как и следует из его названия, завершает сеанс работы с базой (макрокомандой в данном случае служит *Выход* с параметром *Подтверждение*). Полный набор получившихся макросов сравните с рис. 121. Недостающие макросы для открытия форм создаются ана-

логичным образом. Единственным отличием для них является установка режима данных *Изменение*.

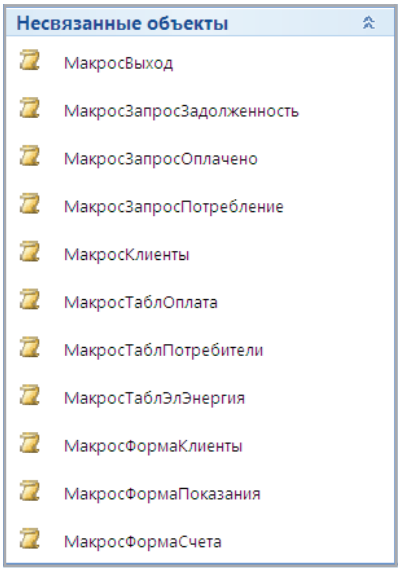


Рис. 121. Список макросов

Размещение элементов на кнопочной форме

Кнопочная форма создается как обычно — кнопкой *Конструктор* формы на закладке *Создание* главного окна базы данных. Автоматическая генерация кнопки на запуск соответствующего макроса выполняется в момент «бросания» макроса, выбранного на закладке Макросы также главного окна базы данных (способ «взял и тащи»).

Кроме кнопок, полученных в результате «бросания» макросов на область данных формы, добавлены такие элементы, как надписи и прямоугольники (рамка). Назначение прямоугольника — сгруппировать родственные по смыслу элементы формы. В данном случае первый прямоугольник окантовывает надпись и кнопки открытия таблиц. Второй и третий прямоугольники выполняют схожие функции для запросов и форм.

Элементы, размещенные на форме, можно вместе перемещать, менять их размеры, выравнивать и т. п. Для этого их следует выделить в общий блок с помощью мышки с удерживанием в нажатом положении клавиши *Shift*. Различный вид прямоугольников достигается с помощью их индивидуальной настройки (*Свойства: оформление*).

Добейтесь, чтобы в окончательном виде кнопочная форма выглядела следующим образом (рис. 122). Сохраните ее под именем «Главная».

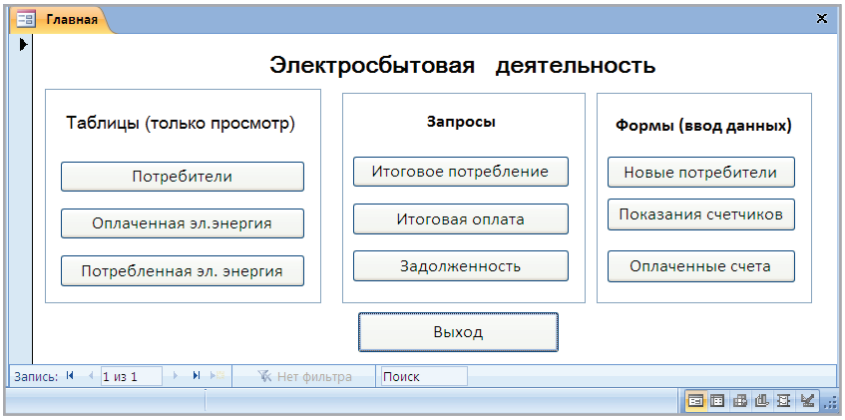


Рис. 122. Форма «Главная»

Настройка последовательностей перехода по форме

Проверьте действие всех кнопок формы. Обратите внимание, в какой последовательности предлагается их нажимать (иначе говоря, «пройдитесь» по форме с помощью клавиш со стрелками клавиатуры или клавишей *Tab*). Почти наверняка очередность обхода кнопок не соответствует логике работы с базой данных. Этот порядок можно заменить с помощью команды **Переходы** в контекстном меню формы в режиме конструктора (рис. 123).

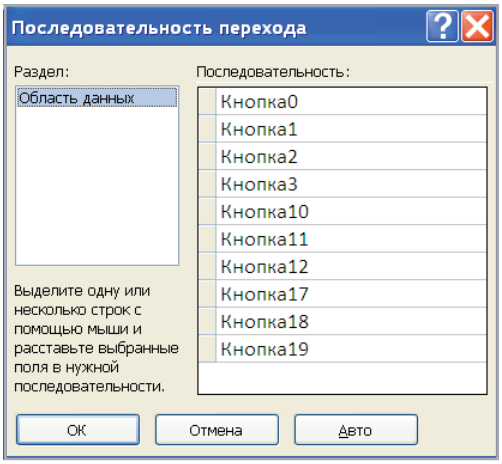


Рис. 123. Определение последовательности перехода

Областью выделения кнопки является квадратик слева от ее имени. Единственное неудобство — здесь кнопки показаны под внутренними именами. Внутреннее имя элемента формы находится в его свойстве *Имя* на закладке *Другие* или *Все*. Установите последовательность переходов сначала по кнопкам для открытия форм (ввод данных), затем по кнопкам для таблиц (контроль данных), а затем — по кнопкам для запросов. Последней в списке очередности должна стать кнопка «Выход».

Этап 6. Настройка параметров запуска базы данных

После того как создана и протестирована кнопочная форма, необходимо установить режим, при котором манипулирование объектами базы в обход этого окна было бы исключено. Иначе говоря, при открытии базы форма «Главная» должна запускаться автоматически. Одним из способов этого является настройка так называемых *Параметров запуска*. Для открытия одноименного окна необходимо из главного окна базы данных нажать кнопку *Office*, затем *Параметры Access* и выбрать раздел *Текущая база данных*. После этого в подразделах *Параметры приложений* и *Параметры ленты* выполнить установки в соответствии с рис. 124.

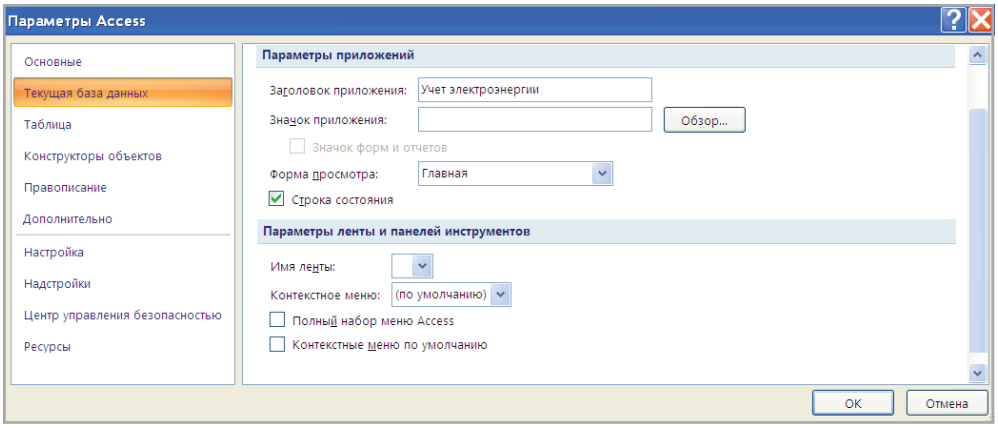


Рис. 124. Настройка параметров запуска базы данных

Поле *Заголовок приложения* определяет фразу, которая будет выводиться в первой строке окна запущенной базы данных. Поле *Форма просмотра* настраивается на кнопочную форму. Отмените полный на-

бор команд Access, чтобы использовался сокращенный, в котором отсутствует доступ к средствам разработки. Также рекомендуется снять опцию *Специальные клавиши Access*, чтобы и здесь ограничить несанкционированные действия пользователя базы.

Запустите базу данных и проверьте ее функционирование. Для возврата к предыдущему режиму работы окно *Параметры Access* можно открыть из контекстного меню любой оставшейся на ленте кнопки.

Этап 7. Окончательное тестирование базы данных

Обратите внимание на следующее. Если для какого-либо абонента отсутствует запись в таблице «Оплаченная электроэнергия» (но имеется в таблице «Потребленная электроэнергия»), результат выполнения запроса «Задолженность» далек от ожидаемого результата. Действительно, в получаемой таблице попросту отсутствуют те потребители, которые израсходовали некоторое количество электроэнергии, но не оплатили его. Причина — в некорректной настройке запроса «Задолженность». Для исправления этого запроса необходимо установить иные параметры объединения запросов (рис. 125).

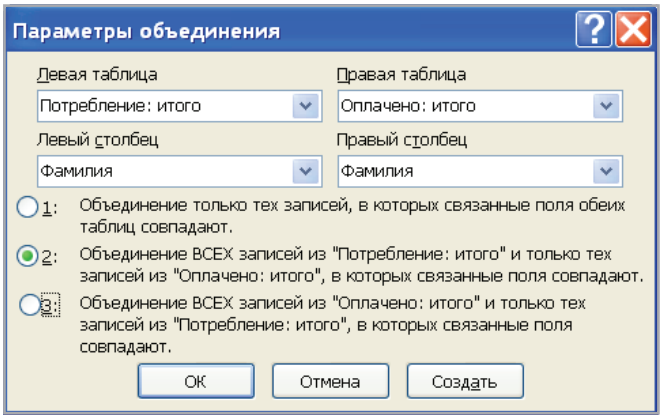


Рис. 125. Параметры объединения запросов

После этого на линии, графически иллюстрирующей связь, появится стрелка, направленная в сторону запроса «Оплачено: итого».

Выполните запрос. В результирующей таблице присутствуют уже все абоненты, потребляющие электроэнергию. Однако сумма задолженности по-прежнему вычисляется неправильно (рис. 126).

Главная | Задолженность

Фамилия	Стоимость	Sum-Сумма оплаты	Задолженность
Иванов	48	26	22
Петров	60	40	20
Семенов	9,2		
Сидоров	72	69	3

Рис. 126. Пример выполнения запроса

Следующий шаг — коррекция формулы, вычисляющей долг. В ней должно быть учтено, что отсутствующее поле в столбце «Sum-Сумма оплаты» представляет собой пустую строку, значением которой является встроенная константа *Null*. В арифметической операции вычитания такое значение использовано быть не может, поэтому результат для первой и третьей записей просто не вычисляется. Решает проблему использование встроенной функции *Ilf* — проверки условия в выражении для вычисления задолженности.

Откройте окно построителя выражений (с помощью [CTRL]/F2) в поле «Задолженность: ...» запроса «Задолженность» в режиме конструктора (рис. 127) и выберите требуемую функцию. В нижней строке построителя выражений Access показывает шаблон для вызова выбранной функции. Здесь *expr* — проверяемое на истинность выражение, truepart — значение, возвращаемое функцией в случае значения испытуемого выражения «Истина», *falsepart* — значение, возвращаемое в случае значения выражения «Ложь». В выражении, вычисляемом задолженность, значение отсутствующего платежа должно быть заменено с помощью функции *IIf* на 0.

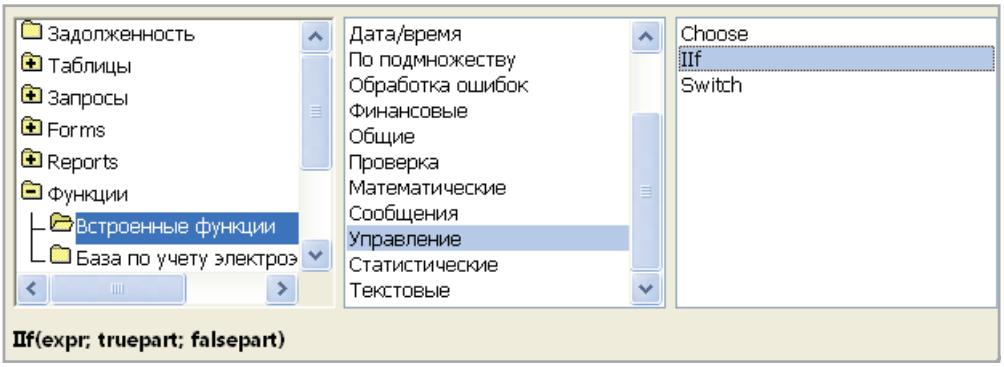


Рис. 127. Выбор встроенной функции

Для этого выполните коррекцию формулы по образцу:

Задолженность: [Потребление: итого]![Стоимость]–
IIf ([Оплачено:итого]![Sum — Сумма оплаты] Is Not Null;
[Оплачено: итого]! [Sum — Сумма оплаты]; 0)

Сохраните запрос и проверьте результат его выполнения. Он должен соответствовать рис. 128.

Главная | Задолженность

Фамилия	Стоимость	Sum-Сумма оплаты	Задолженность
Иванов	48	26	22
Петров	60	40	20
Семенов	9,2		9,2
Сидоров	72	69	3

Рис. 128. Запрос Задолженность

Этап 8. Разработка простого отчета

Отчет — это объект базы данных, специализирующийся на подготовке результирующих данных в виде печатного документа. Отчеты предоставляют широкие возможности для группировки данных и вычислений промежуточных итогов. При этом для каждой группы можно задать отдельный заголовок, определить свое примечание, производить арифметические операции как внутри группы, так и по нескольким группам. Кроме этого, для всего отчета имеется возможность задать примечание, определить верхний и нижний колонтитулы.

Проектируемый отчет предназначается для анализа данных по отпущенной электроэнергии. Построение отчета будем выполнять в режиме конструктора, который включается соответствующей кнопкой в главном окне базы данных на закладке «Создание». Технология создания отчета, по сути, та же, что была при построении форм: на макет перетаскиваются элементы и с помощью свойств настраиваются их параметры.

Настройка свойств отчета и колонтитулов

Сохраните вновь созданный отчет под названием *Потребление за расчетный период*.

Выберите на панели элементов элемент *Надпись*, разместите в области верхнего колонтитула, увеличьте шрифт до 12 пунктов и выберите плотную насыщенность. Самим содержанием надписи может быть строка «Показания электросчетчиков».

В нижнем колонтитуле предусмотрим вывод номера текущей страницы и общего количества страниц. Для этого в области нижнего колонтитула размещается элемент *Поле* и в его свойстве *Данные* задается следующее выражение:

= [Page]&"из"&[Pages]

Свойства *Page* и *Pages* отчета содержат соответственно номер текущей страницы и общее число страниц, строка "из" выводится без изменений, а символ & выполняет «склеивание» строки. Для задания выражения можно воспользоваться *Построителем выражений*.

Присоединенную надпись замените на строку *Страница* и вместе со значением самого поля выводите в отчет курсивом. Результатом первого этапа должен стать макет отчета, подобный рис. 129.

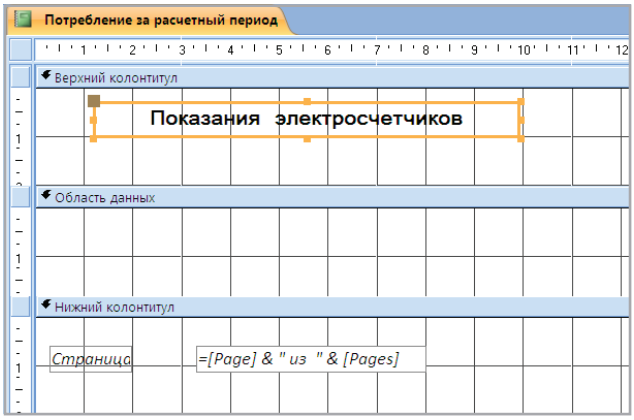


Рис. 129. Макет отчета на первом этапе

Проверить работу отчета можно с помощью команды *Режим* группы *Режимы* на закладке *Главная*.

Построение запроса для отчета

Назначение проектируемого отчета — подготовить в удобном для анализа виде данные о потреблении электроэнергии. Отсюда вывод — источниками должны служить две таблицы: «Потребители» и «Потре-

бленная электроэнергия». Входные данные для отчета определяются в свойстве отчета *Источник записей*. Однако здесь требуется запустить *Построитель запросов* (кнопка с многоточием справа от кнопки раскрывающегося списка). Первым же действием *Построитель запросов* запускает диалог *Добавление таблицы*. Выберите в нем таблицы — источники (рис. 130).

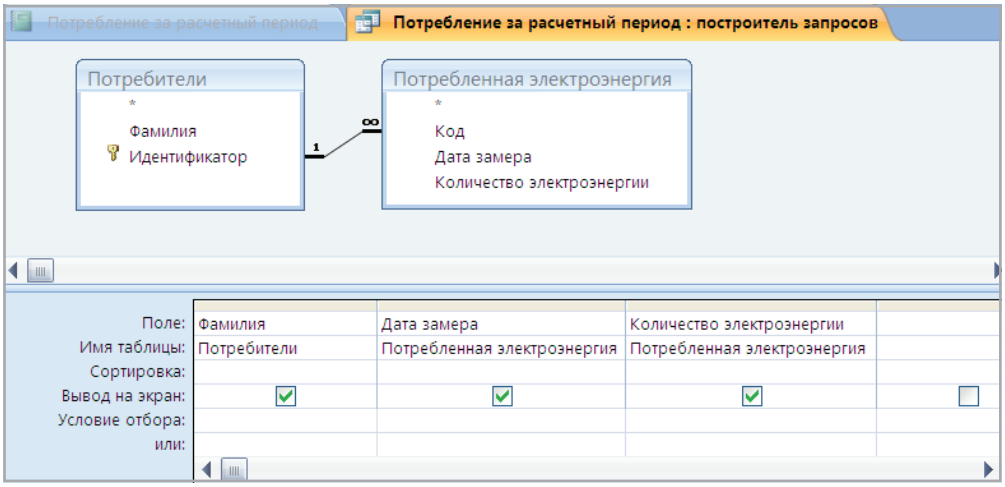


Рис. 130. Построитель запросов

После построения запроса на закладке *Конструктора* следует нажать кнопку *Добавить поля* и отбуксировать поочередно поля запроса, разместив их на макете отчета первоначально по горизонтали. После этого окно *Список полей* можно закрыть. Проверьте работу отчета на данном этапе.

Сортировка и группировка данных в отчете

Результат выполнения отчета пока малопригоден для анализа. Для его улучшения воспользуемся возможностями Access по группировке и сортировке данных. С помощью кнопки *Группировка* на панели инструментов или в контекстном меню отчета открывается соответствующее окно (рис. 131).

Выбором соответствующей кнопки настраивается сортировка по фамилии абонентов, затем группировка по фамилии абонентов с суммированием по полю *Количество электроэнергии*. И, наконец, сортировка по дате замера внутри группы по фамилии. Для группировки *Фами-*

лия необходимо предусмотреть вывод заголовка и примечания группы. После указанных назначений в области данных отчета автоматически выделяются области заголовка и примечания для группы *Фамилия*.

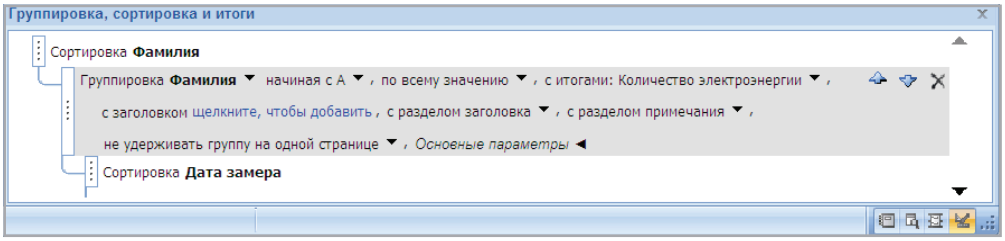


Рис. 131. Окно сортировки и группировки

Следующим шагом переместим в область заголовка группы присоединенные надписи полей данных отчета. Для этого каждую из них сначала нужно отделить от поля командой *Вырезать* меню *Правка*, а затем вставить из буфера на поверхность области заголовка. Проделайте это три раза, а затем увеличьте шрифт надписей, допустим до 11 пунктов, сделайте его плотным и подчеркнутым.

Для элементов макета допустим режим группового изменения одинаковых свойств. Выделите элементы в одну область выделения с помощью мышки, а затем откройте окно *Свойства*. После удаления двоеточия из «бывших» присоединенных надписей вид макета отчета должен быть примерно таким, как на рис. 132. Недостающие надписи в макет отчета можно добавить с помощью элемента *Надпись* панели инструментов. В ответ на замечание Access добавленные надписи могут быть связаны с соответствующим элементом управления. Проверьте отчет в работе.

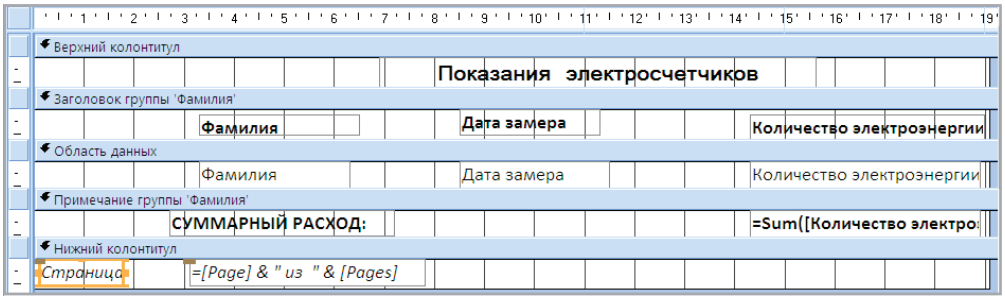


Рис. 132. Второй этап создания макета

В примечании группы добавим среднемесячное потребление электроэнергии. Для этого из панели инструментов на макет перетаскивается элемент Поле, изменяется его надпись, например на строку *Среднемесячное потребление электроэнергии*, а в свойство *Данные* поля заносится следующая формула:

= IIf(Month(Max ([Дата замера]))–Month(Min([Дата замера]))<>0;
Sum([Количество электроэнергии])/
(Month(Max([Дата замера])) — Month(Min([Дата замера]))+1);
Sum([Количество электроэнергии])).

В приведенной формуле используется уже применявшаяся ранее функция *IIf* — выбора одного из двух возможных вариантов. Первый вариант соответствует случаю, когда имеются данные о потреблении за более чем один месяц, т. е. расчетный период не равен нулю. Расчетный период, измеренный в единицах месяцев, находится как разность месяцев между наибольшим и наименьшим месяцами замеров в группе. Для этого используются функции Max() и Min(), возвращающие значения экстремумов и функция Month(), возвращающая номер месяца в дате. Функции выбираются в построителе выражений из раздела Функции → Встроенные функции → Дата/Время.

В итоге условие выбора приобретает вид

Month(Max([Дата замера])) — Month(Min([Дата замера]))<>0.

В этом случае среднемесячное потребление рассчитывается как отношение суммарного расхода к расчетному периоду:

Sum([Количество электроэнергии])/
(Month(Max([Дата замера])) — Month(Min([Дата замера]))+1).

В случае, когда расчетный период, вычисленный в месяцах, нулевой, значением поля становится суммарное количество электроэнергии.

Так как среднемесячное потребление электроэнергии величина с дробной частью, имеет смысл установить формат поля *Фиксированный* с одним или двумя знаками после запятой. Проверьте отчет в действии, выполните необходимую коррекцию в размещении элементов. Уберите повторы фамилий абонентов внутри группы «Фамилия» с по-

мощью соответствующего свойства поля. Окончательный вид отчета должен соответствовать рис. 133. Сохраните отчет под именем «Потребление за расчетный период».

Показания электросчетчиков

Фамилия	Дата замера	Количество электроэнергии
Иванов	31.01.2019	10
Иванов	28.02.2019	30
Иванов	31.03.2019	70
	СУММАРНЫЙ РАСХОД:	**110**
	Среднемесячное потребление электроэнергии:	**36,67**
Фамилия	**Дата замера**	**Количество электроэнергии**
Петров	31.01.2019	20
Петров	28.02.2019	50
Петров	31.03.2019	80
	СУММАРНЫЙ РАСХОД:	**150**
	Среднемесячное потребление электроэнергии:	**50,00**
Фамилия	**Дата замера**	**Количество электроэнергии**
Семенов	19.09.2019	23
	СУММАРНЫЙ РАСХОД:	**23**
	Среднемесячное потребление электроэнергии:	**23,00**
Фамилия	**Дата замера**	**Количество электроэнергии**
Сидоров	31.01.2019	30
Сидоров	28.02.2019	60
Сидоров	31.03.2019	90
	СУММАРНЫЙ РАСХОД:	**180**
	Среднемесячное потребление электроэнергии:	**60,00**

Страница 1 из 1

Рис. 133. Окончательный вид отчета

По аналогии с предыдущим отчетом разработайте отчеты, представляющие в удобном виде данные по платежам за электроэнергию, а также по задолженности потребителей.

14. Основы PowerPoint

Приложение PowerPoint в настоящий момент является не единственным, но самым популярным для работы с презентациями [8]. Настоящее занятие поможет разобраться с основными его инструментами.

Рабочий стол PowerPoint

Документ PowerPoint называется презентацией. Каждая презентация состоит из набора слайдов. Слайд — это минимальный элемент презентации, предназначенный для просмотра на экране или мониторе.

Основную часть окна занимает поле для работы со слайдом, в котором производится редактирование содержимого слайда. Внизу находится поле для заметок к слайду. Данное поле предназначено для данных, которые не будут отображены для зрителей при показе презентации, но при этом будут доступны для рассказчика.

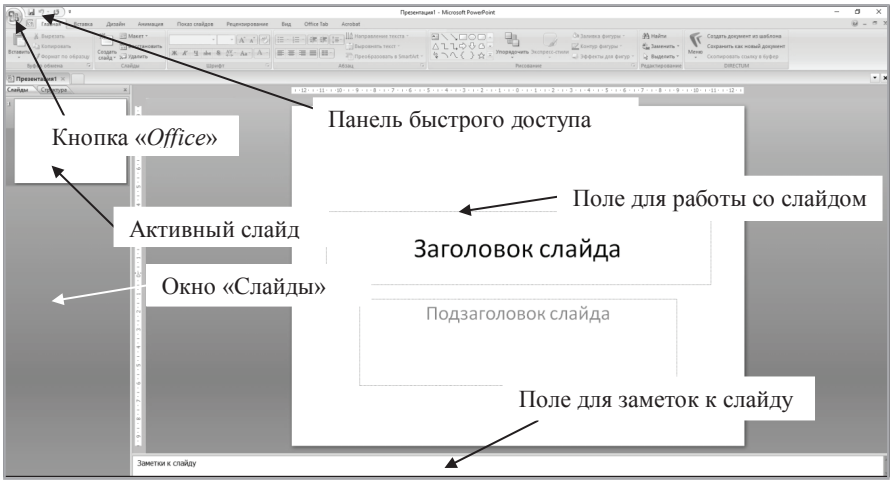


Рис. 134. Рабочий стол PowerPoint

С помощью окна «Слайды» можно как переключиться между слайдами, так и изменить порядок следования слайдов относительно друг друга (по принципу *взял и тащи*).

При создании новой презентации PowerPoint открывает документ, состоящий из одного пустого слайда. Добавление нового слайда возможно путем нажатия кнопки «Создать слайд» в поле «Слайды» вкладки «Главная». Если нажать на нижнюю часть этой кнопки (со стрелочкой), то открывается окно с доступными макетами слайдов (рис. 135). Самый первый слайд презентации по умолчанию создается по макету *Титульный слайд*, второй и последующие — по макету *Заголовок и объект*. Макет любого слайда можно сменить в любой момент разработки презентации. В поле «Слайды» доступны также функции редактирования макетов (кнопка «Макет»), удаления слайда (кнопка «Удалить») и восстановления первоначальных параметров макета (кнопка «Восстановить»).

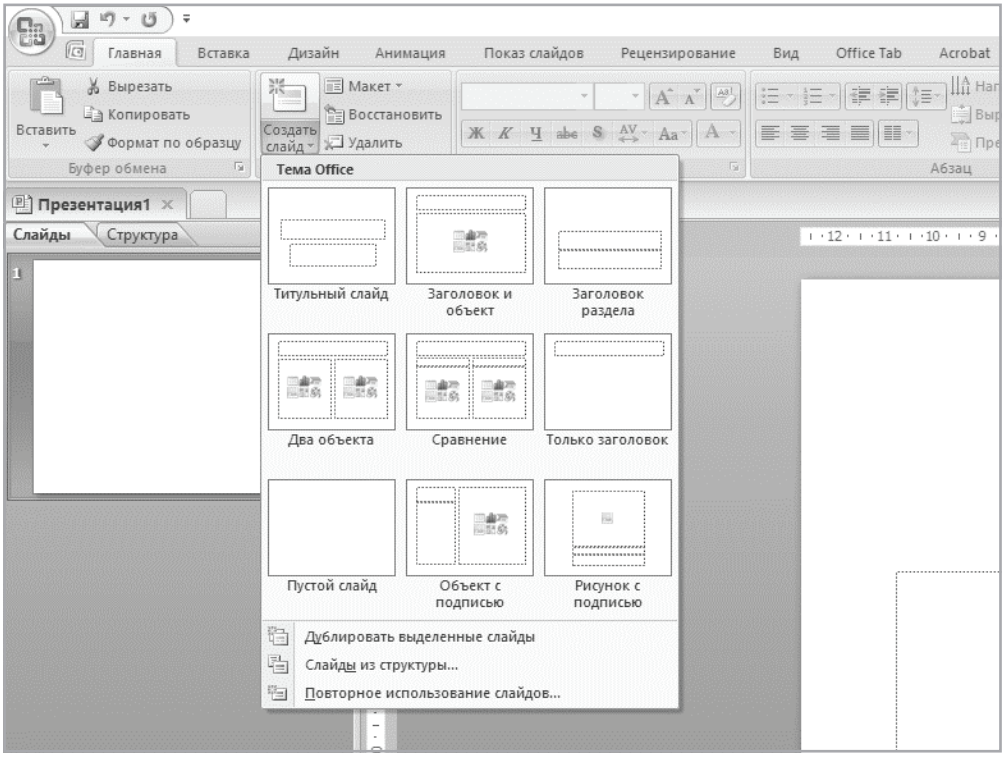


Рис. 135. Макеты создаваемых слайдов

Этап 1. Создание слайдов

Создайте новую презентацию и на первом слайде в поле «Заголовок слайда» запишите «Информатика в энергетике». В поле «Подзаголовок слайда» запишите свою группу, фамилию и инициалы. Шрифт, выравнивание выберите на свое усмотрение. Создайте слайды с макетом «Сравнение» и «Объект с подписью». Удалите поля для ввода заголовков созданных слайдов, сместите поля для ввода текста относительно друг друга. Восстановите прежнее положение объектов на слайдах. Поменяйте порядок следования слайдов. Удалите вновь созданные слайды.

Редактирование содержимого слайда возможно при наличии на слайде редактируемых объектов. На первом слайде в качестве объектов предусмотрены два пустых текстовых поля. В ходе работы можно изменить как их положение (принцип *взял и тащи*), так и содержимое.

Для ввода или редактирования текста необходимо кликом левой клавиши мышки активировать область внутри поля. Область при этом будет выделена пунктирной линией. Ввод и редактирование текста аналогично работе в MS Word.

Для перемещения поля необходимо его выделить, навести курсор мышки на одну из его границ (курсор принимает при этом вид четырехсторонней стрелочки), а затем, нажав и не отпуская левую кнопку мыши, переместить ее в нужную сторону. Подобный принцип используется при изменении границы поля. Единственным отличием является использование специальных меток, расположенных по сторонам и углам изменяемого поля.

Приложение PowerPoint позволяет редактировать не только текст внутри поля, но также и само поле. Основные инструменты для этого сгруппированы в поле «Рисование» вкладки «Главная». Изучите работу указанных инструментов.

Вставка рисунка

При разработке презентации не обойтись без рисунков, графиков, диаграмм других графических объектов.

Для вставки рисунка необходимо перейти на вкладку «Вставка» и в поле «Иллюстрации» нажать кнопку «Рисунок» (рис. 136). Далее в появившемся окне необходимо найти необходимый рисунок, выде-

лить его и нажать кнопку «Вставить». Аналогичный результат достигается уже обсуждаемой ранее комбинацией клавиш Ctrl+C (копировать) и Ctrl+V (вставить). Особенность вставки рисунка с использованием инструмента «Вставка» заключается в том, что при необходимости рисунок автоматически уменьшается до границ слайда. После вставки на рабочем окне активируется вкладка «Формат» для дальнейшей работы с рисунком.

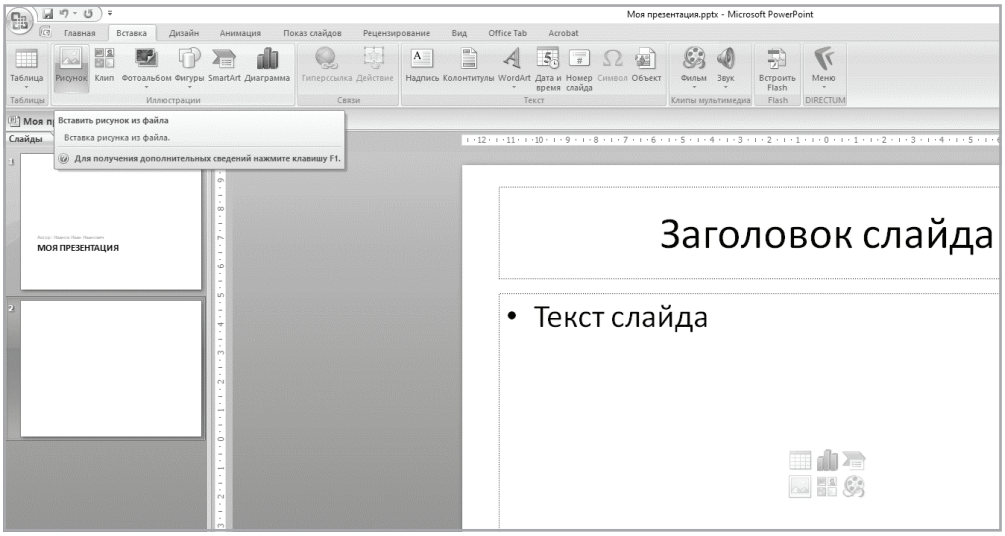


Рис. 136. Вставка рисунка

Аналогичным образом можно вставить иные элементы, такие как «Фигуры», объекты «SmartArt», «WordArt», диаграммы, графики, фильм, звук и так далее. Редактирование вставленных диаграмм и графиков их осуществляется программой Excel.

Этап 2. Вставка рисунка

Добавьте в презентацию новый слайд по макету «Заголовок и объект». В поле «Заголовок слайда» запишите «Закон Ома». В поле Текст слайда запишите «Сила тока в участке цепи прямо пропорциональна напряжению и обратно пропорциональна сопротивлению данного участка цепи» Вставьте рисунок *Закон Ома* из папки P:\metod\Информатика-1\PowerPoint\Рисунки. Под рисунком вставьте надпись «Закон Ома для сети цепи на постоянном токе».

Дизайн слайда презентации

Важным элементом любой презентации является ее художественное оформление. На вкладке *Дизайн* предлагается несколько стандартных тем оформления слайдов. В состав темы входят, как правило, несколько рисунков (часть из которых фоновые), гаммы цветов, шрифты для используемых текстов и другие параметры. Темы являются неким аналогом стилей в Word. Стоит отметить, что дизайн титульного слайда всегда отличается от дизайна остальных слайдов. Изучите возможности выбора стандартных тем самостоятельно.

PowerPoint предоставляет пользователю возможность создания своего стиля, в частности, добавлением рисунков, изменением текстовых шрифтов и других настроек.

Для того чтобы создать свой собственный дизайн презентации, необходимо на вкладке «Вид» в поле «Режимы просмотра презентации» выбрать инструмент «Образец слайдов» (рис. 137). В результате происходит переход на вкладку «Образец слайдов». С левого края будет отображен набор макетов слайдов (слайды поменьше), которые основываются на образце слайда (большой слайд). Все изменения, которые вносятся в образец слайдов, будут автоматически перенесены во все макеты слайдов, но при этом изменение макета одного слайда никак не отразится на остальных макетах слайдов и на образце слайда.

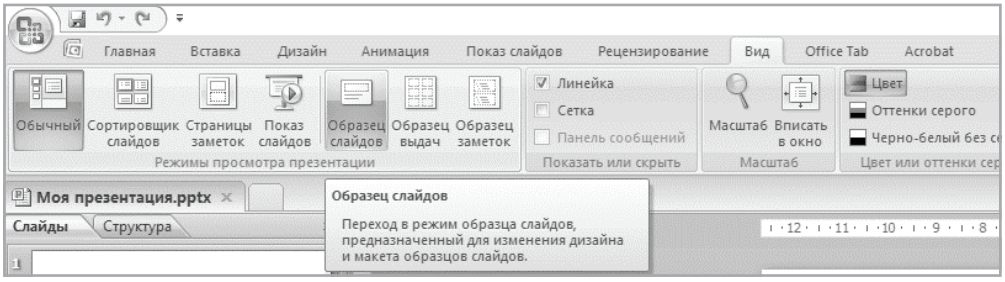


Рис. 137. Выбор инструмента Образец слайдов

В режиме «Образец слайдов» можно редактировать существующие макеты и образец слайдов, а также создавать свои собственные. Для того чтобы создать новый образец слайдов, необходимо в поле «Изменить образец» нажать на кнопку «Вставить образец слайдов». Аналогичным образом можно создать новый макет слайда, нажав на кнопку

«Вставить макет» (рис. 138). Стоит отметить, что при вставке нового образца слайдов автоматически вставляются все стандартные макеты слайдов.

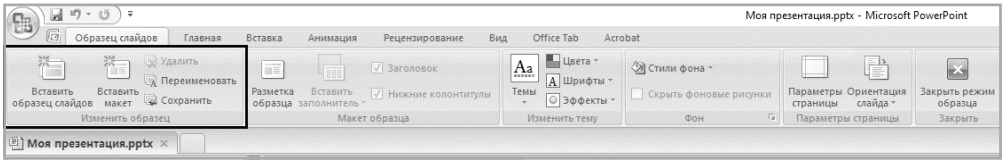


Рис. 138. Поле Изменить образец

При разработке презентации зачастую применяется единый фон для всех слайдов презентации. Для этого на макете слайда, изображенного в левой части рабочего стола, необходимо нажать правую кнопку мыши и в контекстном меню выбрать команду «Формат фона». В появившемся окне можно выбрать несколько способов заливки:

— сплошная заливка — весь фон будет выполнен монотонным цветом;
— градиентная заливка — весь фон будет выполнен в виде перехода от одного цвета к другому (к нескольким цветам) с заданными параметрами;
— рисунок или текстура — фон будет выполнен в виде загруженного рисунка, либо текстуры, предоставленной PowerPoint.

Также в режиме слайдов можно редактировать дизайн тех объектов, которые по умолчанию не видны на слайдах, таких как «Дата и время», «Номер слайда» и «Колонтитулы». Выполняется это после вставки соответствующего объекта с помощью поля «Текст» вкладки «Вставка».

Этап 3. Создание дизайна презентации

В качестве формата фона выберите рисунок «Фон PowerPoint. УрФУ УралЭНИН АЭС» из папки P:\metod\Информатика-1\PowerPoint\Рисунки. Откорректируйте расположение заголовков таким образом, чтобы они не перекрывали часть слайда, занятую рисунком. При необходимости откорректируйте расположение рисунка и надписи в ранее созданном слайде. Сделайте так, чтобы номер слайда и дата создания слайдов отображались на каждом слайде при их создании.

Анимация объектов слайда

Дополнительную наглядность презентации обеспечивают анимированные объекты. Анимация в PowerPoint подразделяется на четыре следующие категории:

— вход. Данный тип анимации позволяет появиться выбранному элементу с соответствующим эффектом;
— выход. Данный тип анимации позволяет исчезнуть выбранному элементу с соответствующим эффектом
— выделение. Данный тип анимации позволяет измениться выбранному элементу с соответствующим эффектом
— пути перемещения. Данный тип анимации позволяет двигаться выбранному элементу по заданной траектории.

Для того чтобы использовать анимацию входа, необходимо перейти на вкладку «Анимация» и поле «Анимация» нажать на кнопку «Настройка анимации». После этого откроется окно «Настройка анимации» (рис. 139).

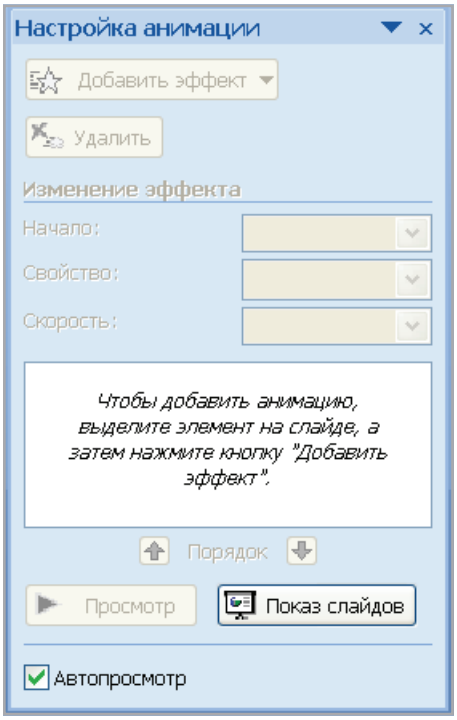


Рис. 139. Настройка анимации

Далее необходимо выбрать объект, который планируется анимировать, нажав на него левой кнопкой мыши. В результате в окне настройки анимации становится активной кнопка «Добавить эффект». После нажатия на нее появится список, из которого необходимо выбрать категорию «Вход», а затем выбрать один из доступных видов анимации (рис. 140).

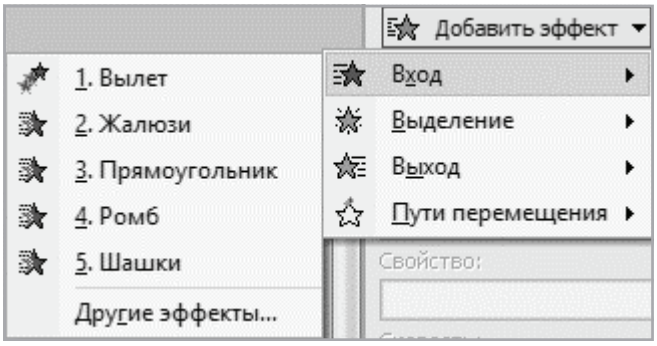


Рис. 140. Выбор анимации категории «Вход»

Чтобы посмотреть все доступные эффекты анимации входа, необходимо нажать на кнопку «Другие эффекты...». Чтобы оценить, как будет работать анимация перед ее выбором, необходимо установить галочку «Просмотр эффекта» (рис. 141). Выбрав желаемый эффект, необходимо нажать кнопку «ОК» и эффект будет добавлен в область анимации. Аналогичным образом можно добавить эффекты выхода и выделения. Выбирая пути перемещения можно выбрать путь как среди стандартных, так и нарисовать собственный путь перемещения (рис. 142).

Если на слайде имеется несколько эффектов анимации, то необходимо определить их последовательность между собой. В области «Настройка анимации» тот эффект, который находится выше, будет выполнен раньше, чем эффект, расположенный ниже.

Одновременно с этим стоит отслеживать также способ запуска эффекта. Его можно запустить следующим образом:

— «Запускать щелчком» — эффект будет запущен только при щелчке кнопки мыши;
— «Запускать вместе с предыдущим» — эффект будет запущен вместе с предыдущим эффектом;
— «Запускать после предыдущего» — эффект будет запущен сразу после окончания предыдущего эффекта.

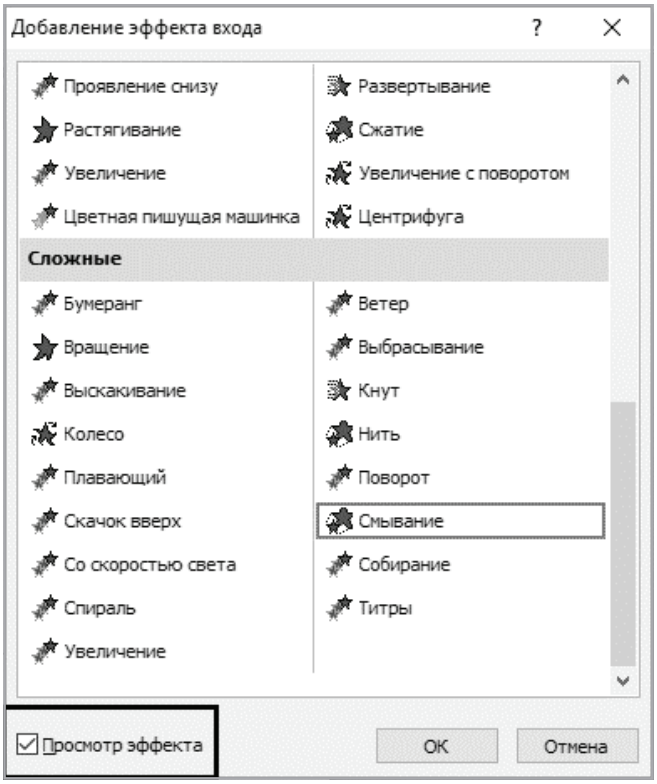


Рис. 141. Добавление эффекта входа с предварительным просмотром

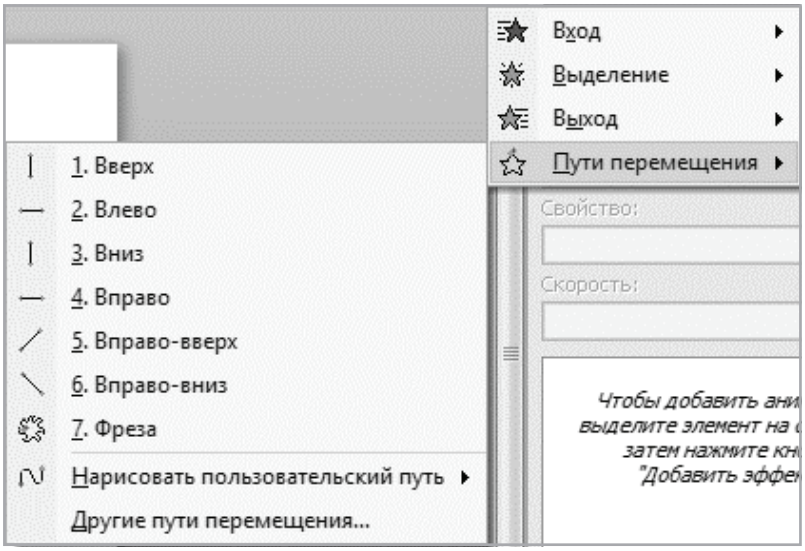


Рис. 142. Выбор анимации категории «Пути перемещения»

Данные параметры можно изменить в окне настройки анимации либо в контекстном меню соответствующего эффекта, либо в списке «Начало:» в области «Изменение: Вылет». В контекстном меню эффекта можно также выбрать пункт «Параметры эффектов» (рис. 143). В появившемся окне во вкладке «Время» присутствует такое поле, как «Повторение». Благодаря настройке данного поля, можно настроить циклическую анимацию, тем самым оживив выбранную картинку.

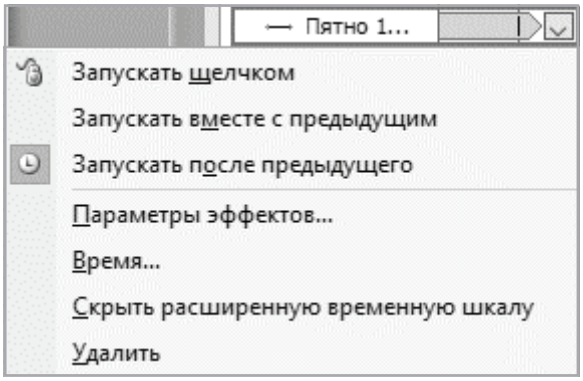


Рис. 143. Контекстное меню эффектов

Самостоятельно изучите общие и индивидуальные параметры анимации 3–5 эффектов каждой группы, опробуйте варианты анимации переходов между слайдами, ответьте (в отчете) на следующие вопросы:

1. Каким образом применить единый переход ко всем слайдам презентации?
2. Как настроить автоматическое переключение слайдов через заданный промежуток времени?
3. Каким образом регулируется скорость перехода между слайдами?

Этап 4. Создание анимации объектов

В качестве индивидуального задания создайте на новом слайде анимированную презентацию, аналогичную примерам, расположенным в папке P:\metod\Информатика-1\PowerPoint\Образцы анимации в соответствии с вариантом, указанным преподавателем:

1. Вращение ротора внутри статора;
2. Перевозка топлива из топливного бассейна на электростанцию;
3. Движение магнитного поля между двумя полюсами магнита;

4. Распределение электроэнергии на подстанции;
5. Последствия прикосновение человека к токоведущим частям;
6. Движение магнитного потока в трансформаторе;
7. Вращение лопастей ветрогенератора как следствие действия ветра;
8. Нагрев теплоносителя на солнечной электростанции;
9. Проявление первого закона Кирхгофа;
10. Срабатывание защиты линии при возникновении короткого замыкания.

Создание гиперссылок

Иногда во время представления презентации необходимо изменить порядок следования слайдов. Удобным инструментом в данном случае являются гиперссылки.

Для того чтобы создать гиперссылку, необходимо сначала создать объект, нажатие на который будет осуществлять переход на определенный слайд или документ. Данным объектом может служить любой вставленный объект: надпись, отдельное слово, фигура, рисунок, SmartArt, WordArt, диаграмма и так далее.

Создайте объект SmartArt, выбрав в качестве заготовки вертикальный маркированный список, в элементах которого будут размещены гиперссылки (рис. 144). Затем создайте 5 слайдов, на которые впоследствии будет происходить переход. Рекомендуется в заголовках указывать наименование слайдов, соответствующее пункту на объекте SmartArt.

Рис. 144. Список слайдов

Для того чтобы создать гиперссылку, необходимо войти в режим редактирования элемента списка. После этого выделяем текст, соответствующий слайду, на который будет перенаправлять ссылка. Затем на вкладке «Вставка» в поле «Связи» выбирается инструмент «Гиперссылка».

В появившемся окне (рис. 145) можно сослаться не только на слайды в презентации, но также на другие презентации, а также и на другие файлы. Тем не менее в данном случае создаем ссылку на слайд из презентации, выбрав слайд, соответствующий надписи, и нажимаем кнопку «ОК».

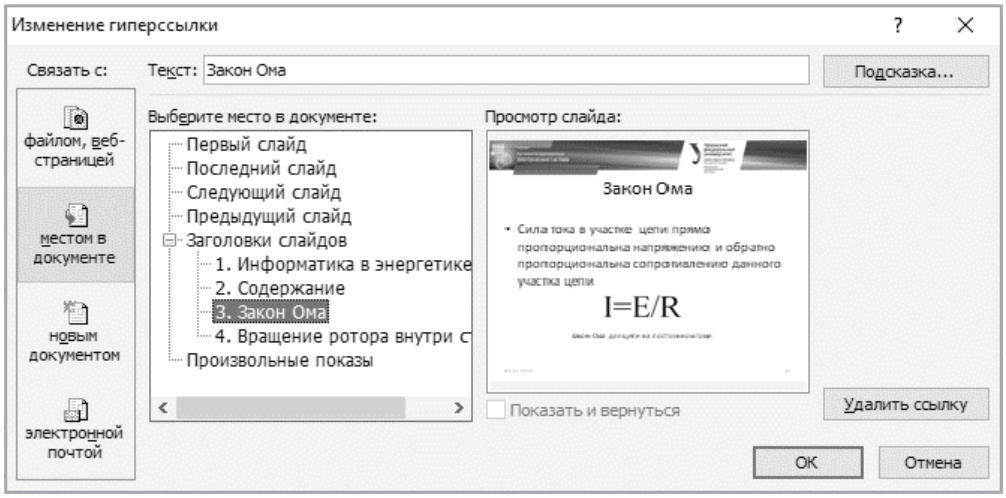


Рис. 145. Изменение гиперссылки

После этого в режиме показа слайдов при нажатии на данный пункт списка произойдет автоматический переход на заданный слайд.

Чтобы удалить ссылку, необходимо войти в окно «Изменение гиперссылки» и нажать на кнопку «Удалить ссылку». Если же ссылка отсутствовала, то кнопка «Удалить ссылку» также будет отсутствовать. Стоит отметить, что для изменения ссылки нет необходимости удаляться старую ссылку.

Этап 5. Создание гиперссылок

Создайте слайд по макету «Заголовок и объект» и установите его вторым по порядку следования. В указанном слайде в поле «Заголо-

вок» запишите «Содержание». В поле «Текст слайда» создайте объект «SmartArt» в формате списка, в котором перечислите заголовки всех имеющихся слайдов за исключением титульного слайда и слайда с содержанием. При необходимости добавьте новые поля в списке или удалите старые поля в списке. Создайте гиперссылки на все имеющиеся слайды за исключением титульного слайда и слайда с содержанием. На каждом слайде определите возможность переключения на любой слайд с помощью гиперссылок на различных объектах.

Показ слайдов

PowerPoint имеет несколько инструментов, которые позволяют достаточно гибко настроить режим показа слайдов подготовленной презентации.

Чтобы начать показ, необходимо перейти на вкладку «Показ слайдов» и в поле «Начать показ слайдов» нажать на одну из следующих кнопок:

— «С начала» — показ слайдов начнется с первого слайда презентации;
— «С текущего слайда» — показ слайдов начнется с того слайда, который отображается в поле редактирования;
— «Произвольный показ» — в данном режиме показ слайдов производится любым удобным для пользователя порядке. Возможен показ одного и того же слайда несколько раз за время презентации.

Для организации произвольного показа сладов в окне «Произвольный показ» необходимо нажать на кнопку «Создать». Порядок слайдов задается пользователем в окне «Задание произвольного показа» (рис. 146). Для того чтобы добавить конкретный слайд в произвольный показ, необходимо выделить его в окне «Слайды презентации» и нажать на кнопку «Добавить».

Этап 6. разработка режима демонстрации слайдов

Создайте произвольный показ слайдов в следующей последовательности:

1. Слайд № 1.
2. Слайд № 3.
3. Слайд № 4.

4. Слайд № 3.
5. Слайд № 2.
6. Слайд № 3.
7. Слайд № 4.
8. Слайд № 1.

При отсутствии необходимого количества слайдов создайте новые слайды. Продемонстрируйте результат Вашей работы преподавателю.

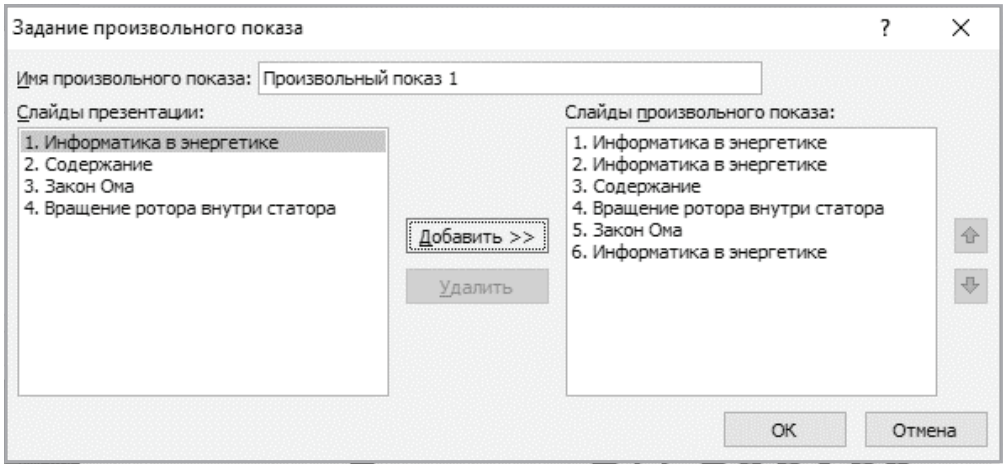


Рис. 146. Окно «Задание произвольного показа слайдов»

PowerPoint позволяет также определить: какие слайды стоит показывать зрителям, а какие — не стоит. Для этого выбрать слайд, который является нежелательным, и в поле «Настройка» вкладки «Показ слайдов» нажать на кнопку «Скрыть слайд». Проверьте работу данного инструмента.

Для того чтобы слайды отображались корректно на всех устройствах, рекомендуется сохранять файл презентации в формат демонстрации (.ppsx,.pps). Данный формат не позволяет выполнять редактирование содержимого презентации, но предоставляет некоторые возможности по управлению просмотром. Меню управления демонстрацией появляется на экране, если в режиме показа слайдов переместить курсор в левый нижний угол экрана (рис. 147).

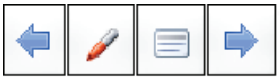


Рис. 147. Управление демонстрацией

Назначение кнопок (слева направо):

— стрелка влево — выполняет переход к предыдущему действию: предыдущему слайду, предыдущему эффекту анимации;
— ручка — редактирование демонстрации. При нажатии на данную кнопку появится меню, в котором можно отредактировать средство для редактирования демонстрации: фломастер, выделение, ручка. Также можно изменить цвет вышеперечисленных средств;
— слайд — навигационная панель. При нажатии на данную кнопку появится меню, с помощью которого можно перейти к любому слайду презентации, завершить демонстрацию. Команда «Справка» выводит окно по быстрым клавишам для управления навигацией (рис. 148).
— стрелка вправо — выполняет переход к следующему действию: к следующему слайду, к следующему эффекту анимации;

Справка по показу слайдов

Во время показа слайдов: OK

Щелчок левой кнопкой, пробел, стрелка вправо или стрелка вниз, клавиша "N", ввода или "Page Down"	Переход к следующему слайду
Клавиша "P", "Backspace", "Page Up", стрелка влево или стрелка вверх	Возврат к предыдущему слайду
Число, затем клавиша ввода	Переход к указанному слайду
"B" или "."	Затемнение и восстановление экрана
"W" или ","	Засветление и восстановление экрана
"A" или "="	Отображение и скрытие указателя
"S" или "+"	Остановка и возобновление показа
"Esc", "Ctrl"+"Break" или "-"	Завершение показа слайдов
"E"	Очистка экрана
"H"	Переход к следующему слайду (если он является скрытым)
"T"	Установка нового времени
"O"	Установка исходного времени
"M"	Настройка времени - продвижение по щелчку мыши
Удержание обеих кнопок мыши в течение 2 секунд	Возврат к первому слайду
Ctrl+P	Замена указателя карандашом
Ctrl+A	Замена указателя стрелкой
Ctrl+E	Замена указателя ластиком
Ctrl+H	Скрытие указателя и кнопки
Ctrl+U	Автоматическое отображение или скрытие указателя
Щелчок правой кнопкой	Меню или предыдущий слайд
Ctrl+S	Диалоговое окно "Все слайды"
Ctrl+T	Панель задач просмотра
Ctrl+M	Отображение или скрытие рукописных примечаний

Рис. 148. Быстрые клавиши для управления навигацией

Изучите средства меню управления демонстрацией и возможности быстрой навигации между слайдами.

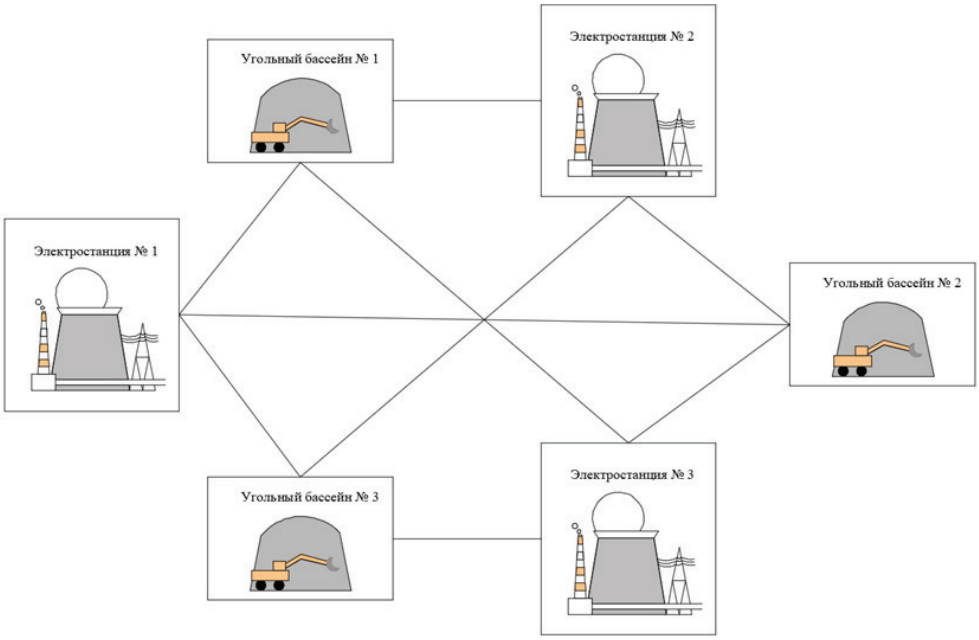
Приложение 2 содержит некоторые общие советы, которых обычно придерживаются при разработке презентаций. Следование данным рекомендациям позволит Вашим презентациям стать более качественными и профессиональными.

Библиографический список

1. Симонович, С. В. Информатика. Базовый курс: учебник для вузов. — 3-е изд. — С. В. Симонович ; Мин-во науки и высшего образования РФ. — Санкт-Петербург : Питер, 2011. — 640 с. ISBN 978-5-459-00439-7.
2. Симонович, С. В. Эффективный справочник. Word, Excel и другие офисные средства / С. В. Симонович, Г. А. Евсеев ; Мин-во науки и высшего образования РФ. — Москва : ОЛМА Медиа Групп, 2006. — 416 с. ISBN 5-373-00078-7.
3. Васильев, А. Н. Excel 2010 на примерах / А. Н. Васильев ; Мин-во науки и высшего образования РФ. — Санкт-Петербург : БХВ-Петербург, 2010. — 432 с. ISBN 978-5-9775-0578-9.
4. Дьяконов, В. П. MathCAD 8 PRO в математике, физике и Internet / В. П. Дьяконов, И. В. Абраменкова ; Мин-во науки и высшего образования РФ. — Москва : Нолидж, 2000. — 512 с. ISBN 5-89251-068-9.
5. Ханова, А. А. Введение в систему MathCAD / А. А. Ханова ; Мин-во науки и высшего образования РФ. — Астрахань : Изд-во АГТУ, 2001. — 33 с.
6. Обоскалов, В. П. Система управления базами данных Access : методические указания / В. П. Обоскалов ; Мин-во науки и высшего образования РФ. — Екатеринбург : УГТУ–УПИ, 2000.– 81 с.
7. Сеннов, А. С. Access 2010. Учебный курс / А. С. Сеннов ; Мин-во науки и высшего образования РФ. — Москва — Санкт-Петербург — Нижний Новгород : Питер, 2010. — 288 с. ISBN 978-5-49807-806-9.
8. Шульгин, В. П. Создание эффективных презентаций с использованием PowerPoint 2013 и других программ / В. П. Шульгин, М. В. Финков, Р. Г. Прокди ; Мин-во науки и высшего образования РФ. — Санкт-Петербург : Наука и Техника, 2015. — 256 с. ISBN 2227-8397.

Приложение 1

Варианты заданий для транспортной задачи



Вариант А

Таблица П1.1

Параметры электрических станций

№	Установленная мощность, $P_{УСТ}$, МВт			Число часов выработки установленной мощности, $T_{УСТ}$, час			Удельный расход станции, г. у. т./кВт·час		
	1	2	3	1	2	3	1	2	3
1	200	100	150	5000	5500	6000	303	310	313
2	120	100	150	5500	5800	5000	304	311	314
3	200	300	150	5500	5800	5000	305	312	315
4	60	300	200	6500	5000	4500	306	313	316
5	80	100	300	6000	5500	4500	307	314	317
6	320	160	200	5000	6800	5000	308	315	318
7	30	400	300	7000	5500	5200	309	316	310
8	50	100	500	7000	5500	4500	310	317	311

Окончание табл. П1.1

№	Установленная мощность, $P_{УСТ}$, МВт			Число часов выработки установленной мощности, $T_{УСТ}$, час			Удельный расход станции, г. у. т./кВт·час		
	1	2	3	1	2	3	1	2	3
9	300	400	120	4500	4000	6200	311	318	312
10	300	500	150	5500	4000	5000	312	303	304
11	80	1200	50	5500	4500	5000	313	304	305
12	60	200	1000	6500	4800	4000	314	305	306
13	40	300	500	6000	5500	4000	315	306	307
14	60	80	1200	500	6800	5000	316	307	308
15	240	300	900	7000	5500	5200	317	308	309

Таблица П1.2

Параметры угольных бассейнов

№	Годовая добыча топлива в угольных бассейнах, тыс. тонн			Цена топлива в угольном бассейне, у. е./тонна		
	1	2	3	1	2	3
1	300	200	600	1,5	3,5	2
2	200	200	600	1,5	3,5	2
3	700	200	600	1,5	1,8	2
4	700	200	300	2,5	2,4	1,7
5	100	600	300	2,5	2,6	1,7
6	700	500	300	2,5	2,8	1,7
7	700	500	400	3,6	3,2	3,4
8	300	600	400	3,6	2,2	3,4
9	700	500	300	3,6	2,2	3,4
10	400	500	1300	3,6	2,2	3,4
11	1500	800	400	3,5	4,3	4,2
12	900	200	1200	1,2	4,2	3,5
13	700	600	300	1,2	4,1	3,4
14	1900	500	600	1,2	4,0	3,3
15	1200	1700	500	2,2	3,9	3,2

Таблица П1.3

Параметры перевозки топлива

№	Расстояние от угольного бассейна до станции, км									Стоимость перевозки, у. е./тонна·км
	1–1	1–2	1–3	2–1	2–2	2–3	3–1	3–2	3–3	
1	50		100	44	20	90	70	120	160	0,07
2	70	120	160		20	90	20	120	165	0,065
3	40	120	80	44		150	20	120	170	0,06
4	20	40	80	60	40		20	200	175	0,055
5		150	20	200	160	150	25	26	35	0,055
6	20	190		200	160	150	79	26	70	0,02
7	20	150	50	200	160		25	26	100	0,03
8	20	80	50	120		150	25	120	110	0,04
9	60	200	11	120	160	210	44	20	90	0,05
10		120	160	50	20	90	60	120	280	0,01
11	40	120		44	20	150	40	120	170	0,05
12	70	50	40	60	55	150	90	200	175	0,045
13	40	150	20		160	150	35	47	45	0,01
14	75	190	50	200		180	79	65	77	0,02
15	50	150	30	200	160	150		65	129	0,03

Вариант Б

Таблица П1.4

Параметры электрических станций

№	Установленная мощность, $P_{УСТ}$, МВт			Число часов выработки установленной мощности, $T_{УСТ}$, час			Удельный расход станции, г. у. т./кВт·час		
	1	2	3	1	2	3	1	2	3
1	80	100	300	5000	5500	6000	308	315	313
2	320	160	200	5500	5800	5000	309	316	315
3	30	400	300	5500	5800	5000	310	317	316
4	50	100	500	6500	5000	4500	311	318	317
5	300	400	120	6000	5500	4500	312	303	318
6	300	500	150	5000	6800	5000	313	304	303
7	80	1200	50	7000	5500	5200	314	305	304
8	60	200	1000	7000	5500	4500	315	306	311
9	40	300	500	4500	4000	6200	316	318	312
10	60	80	1200	5500	4000	5000	317	303	304

Окончание табл. П1.4

№	Установленная мощность, $P_{УСТ}$, МВт			Число часов выработки установленной мощности, $T_{УСТ}$, час			Удельный расход станции, г. у. т./кВт·час		
	1	2	3	1	2	3	1	2	3
11	240	300	900	5500	4500	5000	318	304	305
12	400	1200	150	6500	4800	4000	308	305	306
13	200	800	200	6000	5500	4000	315	306	307
14	300	2000	80	500	6800	5000	316	307	308
15	90	250	1600	7000	5500	5200	317	308	309

Таблица П1.5

Параметры угольных бассейнов

№	Годовая добыча топлива в угольных бассейнах, тыс. тонн			Цена топлива в угольном бассейне, у. е./тонна		
	1	2	3	1	2	3
1	300	200	600	1,15	3,50	2,00
2	200	200	600	1,50	3,50	2,00
3	700	200	600	1,50	3,50	2,00
4	700	200	300	2,53	1,22	1,70
5	100	600	300	2,50	1,20	1,70
6	700	500	300	2,50	1,21	1,37
7	700	500	400	3,60	2,20	3,40
8	300	600	400	3,46	2,20	3,40
9	700	500	300	3,60	2,20	3,40
10	400	500	1300	3,60	2,23	3,40
11	1500	800	400	1,80	4,30	4,52
12	900	200	1200	2,40	4,24	3,50
13	700	600	300	2,60	4,19	3,40
14	1900	500	600	2,80	4,00	3,30
15	1200	1700	500	3,20	3,90	3,21

Таблица П1.6

Параметры перевозки топлива

№	Расстояние от угольного бассейна до станции, км									Стоимость перевозки, у. е./тонна·км
	1–1	1–2	1–3	2–1	2–2	2–3	3–1	3–2	3–3	
1	50	30	100	44	20	90		120	160	0,01
2	70	120	160	20	90	20		120	165	0,02
3	40		120	80	44	150	20	120	170	0,03
4	20	40	80		60	40	20	200	175	0,04
5	20	150	20	200		150	25	26	35	0,05
6	20	190	200		160	150	79	26	70	0,06
7	20	150	50	200	160		25	26	100	0,07
8	20	80	50	120		150	25	120	110	0,07
9	60		210	120	160	150	44	20	90	0,065
10	120	160	50		20	90	60	120	280	0,06
11	40	120		44	20	150	40	120	170	0,055
12	70	50	40	60	55		90	200	175	0,05
13	40		150	20	160	150	35	47	45	0,045
14	75	190	50	200		180	79	65	77	0,02
15	50	150	30	200	160		150	65	129	0,03

Вариант В

Таблица П1.7

Параметры электрических станций

№	Установленная мощность, $P_{УСТ}$, МВт			Число часов выработки установленной мощности, $T_{УСТ}$, час			Удельный расход станции, г. у. т./кВт·час		
	1	2	3	1	2	3	1	2	3
1	240	100	150	5000	5600	6000	312	310	310
2	160	100	150	5500	5900	5000	313	311	311
3	200	360	150	5500	5900	5000	314	312	312
4	60	380	200	6500	5100	4500	315	304	304
5	80	110	320	6000	6500	4500	316	305	305
6	320	160	220	5000	6500	5000	317	306	306
7	30	400	320	7000	5600	5200	318	307	307
8	50	110	500	7000	5600	4500	303	308	308
9	300	400	160	4500	4500	6200	304	309	312
10	300	500	183	5500	4500	5000	305	303	304

Окончание табл. П1.7

№	Установленная мощность, $P_{УСТ}$, МВт			Число часов выработки установленной мощности, $T_{УСТ}$, час			Удельный расход станции, г. у. т./кВт·час		
	1	2	3	1	2	3	1	2	3
11	80	1200	64	5500	4700	5000	312	304	305
12	64	200	1000	6500	4500	4000	313	305	306
13	40	340	500	6000	5000	4000	314	306	307
14	60	85	1200	500	6600	5000	316	307	308
15	240	300	960	7000	5700	5200	317	308	309

Таблица П1.8

Параметры угольных бассейнов

№	Годовая добыча топлива в угольных бассейнах, тыс. тонн			Цена топлива в угольном бассейне, у. е./тонна		
	1	2	3	1	2	3
1	100	600	300	2,5	1,2	3,2
2	700	500	300	2,5	1,2	3,5
3	700	500	400	3,6	2,2	1,2
4	300	600	400	3,6	2,2	3,4
5	700	500	300	3,6	2,2	3,4
6	400	500	1300	3,6	2,2	3,4
7	1500	800	400	1,8	4,3	4,2
8	900	200	1200	2,4	4,2	2
9	700	600	300	2,6	4,1	2
10	1900	500	600	2,8	4,0	1,7
11	1200	1700	500	3,2	3,9	1,7
12	1700	100	900	2,2	2,5	1,7
13	1400	200	1300	2,2	3,6	3,4
14	1500	1800	1200	2,2	3,6	1,2
15	900	100	1500	4,3	3,6	1,2

Таблица П1.9

Параметры перевозки топлива

№	Расстояние от угольного бассейна до станции, км									Стоимость перевозки, у. е./тонна·км
	1–1	1–2	1–3	2–1	2–2	2–3	3–1	3–2	3–3	
1		150	40	200	160	150	25	26	35	0,19
2	20	190		160	200	150	79	26	70	0,2
3	20	150	50	200	160	25	26	100		0,21
4	20	80		50	120	150	25	120	110	0,22
5	60		210	120	80	150	44	20	90	0,23
6	120	160	50		20	90	60	120	280	0,24
7	40	44	120		20	150	40	120	170	0,25
8	70	50	50	60	55	150	90		175	0,26
9	40	150		20	160	150	35	47	45	0,27
10	75	190	50	200	180	79	65	77		0,28
11		50	150	30	200	150	160	65	129	0,275
12	50	30	100	23	92	135		110	170	0,27
13	70	115	165	85	24		94	112	280	0,265
14	40	115	125	44	142	25		114	140	0,26
15	70	155	63	45	145	36	116		150	0,255

Приложение 2

Общие рекомендации по разработке презентаций

Презентация является только инструментом передачи информации. Источником информации в первую очередь является сам рассказчик, а качество презентации определяется, прежде всего, глубиной проработки исходного материала.

Схема презентации

Любая презентация должна иметь минимальный набор слайдов:

1. Титульный слайд — на данном слайде указывается тема работы, имя автора работы, руководителя, рецензента (если таковые имеются), дата и организация, от которой была выполнена работа (университет, институт, кафедра, специальность), дисциплина/модуль/траектория, по которой была подготовлена презентация;
2. Цели и задачи работы;
3. Содержание — основные разделы и работы. Для большей интерактивности можно оформить каждый пункт в виде гиперссылки. Стоит отметить, что цвет гиперссылок до и после использования меняется, поэтому необходимо проверить на сочетание текста гиперссылок и фона до и после их использования;
4. Основные слайды, в которых раскрывается тема;
5. Выводы по проделанной работе;
6. Список использованных источников;
7. Благодарность аудитории за внимание. На данном слайде можно показать свои контакты, а также выразить благодарность всем, кто принимал участие в создании данной работы.

Для лучшего восприятия информации между перелистыванием слайдов рекомендуется давать небольшую паузу. Таким образом, слушатель получает возможность завершить осмысление полученной информации и повысить готовность воспринимать новую порцию данных.

Требования к оформлению слайдов

Необходимо максимально полностью использовать все доступное для слайда пространство. При этом следует учитывать, что в ряде случаев для задних рядов слушателей нижняя часть презентации может ока-

заться зрительно недоступна. По этой причине рекомендуется использовать верхнюю и среднюю часть слайда (ориентировочно 75–80 %).

Рекомендованный размер шрифта варьируется в пределах от 18 до 36 для основного текста и от 24 до 54 для заголовков слайдов. Меньший размер шрифта является проблематичным для рассмотрения, слишком большой размер шрифта не позволяет поместить на слайд всю необходимую информацию. При этом обычно шрифт в течение всей презентации сохраняет свои параметры. Для использования в презентациях рекомендуется применять следующий набор шрифтов: Verdana, Arial, Tahoma, Times New Roman, Georgia, Cambria, Calibri. В любом случае, не стоит использовать шрифты, которые не устанавливаются по умолчанию вместе с операционной системой.

Стоит отметить, заголовок должен присутствовать на каждом слайде и, как правило, свой собственный. Если же не удается снабдить два следующих друг за другом слайда разными заголовками, допускается использовать на втором слайде такой же заголовок, как на первом, но с пометкой «Продолжение» («Продолжение 1», «Продолжение 2» и т. д.).

Также рекомендуется при создании презентации придерживаться правила «не больше трех»: не больше трех фактов на слайде, не больше трех цветов на слайде, не больше трех объектов на слайде, не больше трех минут времени комментариев по одному слайду. Информация, представленная на слайде, должна состоять из максимально емких слов и предложений. Для выделения на слайде более важной информации рекомендуются следующие способы: подчеркивание, курсив, полужирное начертание, выделение цветом, регистром. При этом выделенная информация не должна занимать больше 30 % от общей информации, представленной на слайде.

Цветовая гамма

При создании индивидуального цветового стиля необходимо учитывать восприятие человеком определенных сочетаний цветов. Основной текст и фон должны находиться в контрасте по отношению друг к другу, но при этом не раздражать зрение. Не рекомендуется сочетание зеленого и красного цветов. Предпочтительны следующие сочетания:

— желтый текст на синем фоне;
— белый текст на темно-синем фоне;
— черный текст на белом фоне.

Фон может представлять из себя монотонный цвет, градиентный переход или некое изображение.

Оформление дополнительных материалов

К дополнительным материалам относят графики, рисунки, формулы, объекты WordArt и SmartArt. Их использование значительно повышает восприимчивость презентации. При использовании графиков/рисунков/формул следует предусмотреть поясняющие надписи, которые, как правило, располагаются под объектами. При использовании дополнительных материалов не стоит забывать уже упомянутое правило «не больше трех» и то, что один графический объект не должен занимать более половины слайда. Если на слайде представлена только одна формула/график/рисунок, то вместо поясняющей надписи можно использовать заголовок слайда. Заимствованные материалы слайда должны быть снабжены соответствующей ссылкой.

Анимация

Анимация, использованная в презентации, должна помогать понять материал, а не отвлекать на себя внимание зрителей. Соответственно, не рекомендуются слишком сложные виды анимации. Оптимальным является использование анимации, которая сопровождает изложение материала. Также анимация рекомендуется при объяснении динамических процессов (вращение магнитного поля ротора относительно статора либо наоборот, движение автомобиля, изменение положения коммутационного оборудования на схемах и т. д.). Зачастую при этом используется цикличная анимация, при которой объекты постоянно повторяют одни и те же движения.

Образец презентации с примерами корректного и некорректного представления информации располагается в папке P:\metod\Информатика-1\PowerPoint\

Учебное издание

Котов Олег Михайлович
Котова Елена Николаевна
Верхозин Андрей Михайлович

**ОСНОВЫ ПРЕДСТАВЛЕНИЯ
И ОБРАБОТКИ ДАННЫХ
В ЦИФРОВЫХ СИСТЕМАХ**

Редактор О. С. Смирнова
Верстка О. П. Игнатьевой

Подписано в печать 02.07.2020. Формат 70×100/16.
Бумага офсетная. Цифровая печать. Усл. печ. л. 16,8.
Уч.-изд. л. 9,9. Тираж 100 экз. Заказ 44.

Издательство Уральского университета
Редакционно-издательский отдел ИПЦ УрФУ
620049, Екатеринбург, ул. С. Ковалевской, 5
Тел.: +7 (343) 375-48-25, 375-46-85, 374-19-41
E-mail: rio@urfu.ru

Отпечатано в Издательско-полиграфическом центре УрФУ
620083, Екатеринбург, ул. Тургенева, 4
Тел.: +7 (343) 358-93-06, 350-58-20, 350-90-13
Факс: +7 (343) 358-93-06
http://print.urfu.ru

ИЗДАНИЕ УНИВЕРСИТЕТСКОГО КНИГОИЗДАНИЯ
УНИВЕРСИТЕТСКАЯ КНИГА
2018

9 785799 630126